

Appendix B. Software Libraries

This appendix lists all ECS libraries in alphabetical order. Each library name is followed by a directory specification indicating where that process was defined. A brief description of the library may be given. This is followed by a list of classes and templates used by the library, if any, which are defined in that same directory. These libraries are linked into processes listed in Appendix A.

Some libraries have both static and shared versions. The shared libraries will be indicated by “(shared)” in the heading. In all cases, the name of a shared library ends in “Sh”. However, some static libraries also have names that end in “Sh”.

It should be noted that this appendix contains libraries that are in the Drop 6A code as well as libraries which are in later Drops but may not be fully operative until the delivery of the future Drops.

The first sentence of a class description is taken from the comments, if any, in the code for that class.

library agentevent

/ecs/formal/CSS/DOF/src/LOGGING/agentlog

Purpose of this library is to provide a framework for MSS Agent to store information to the MSS log file and rename the file when the file is full.

Classes

- EcUtAgentLogger -- This class is a subclass of EcUtLoggerRelA.

library agenteventSh (shared)

/ecs/formal/CSS/DOF/src/LOGGING/agentlog

Purpose of this library is to provide a framework for MSS Agent to store information to the MSS log file and rename the file when the file is full.

Classes

- EcUtAgentLogger -- This class is a subclass of EcUtLoggerRelA.

library CIDtProfileToOdl

/ecs/formal/CLS/DESKT/src/OdlProfile

This library is used to read and write ODL parameters specific for the User Profile Gateway.

Classes

- CIDtDataValue
- CIDtDateTime
- CIDtODLAggregate -- This class extracts aggregate information from input aggregate ODL tree.
- CIDtODLNonAggregate -- This class extracts non aggregate information from input aggregate ODL tree.
- CIDtODLParameter
- CIDtODLTree
- CIDtRSTAggregate
- CIDtRSTNonAggregate

library ClOdCommon

/ecs/formal/CLS/ODFRM/src/Common

Classes

- ClOdBase
- ClOdCgiVars -- ClOdCgiVars class to parse arguments from a Web server for use by a cgi-bin program.
- ClOdGIFromODL -- This class builds/gets root node of an input ODL Tree.
- ClOdODLFromGI -- This class builds/gets root node of an input ODL Tree.

library ClWbJtdb

/ecs/formal/CLS/WKBCH/ClWbMp/db

library ClWbJtFolioUtils

/ecs/formal/CLS/WKBCH/ClWbMp/foliod

library ClWbJtgd

/ecs/formal/CLS/WKBCH/ClWbMp/gd

library ClWbJtMap

/ecs/formal/CLS/WKBCH/ClWbMp/maplib

library ClWbJtProjections

/ecs/formal/CLS/WKBCH/ClWbMp/libproj

library Common

/ecs/formal/DM/GatewayCSCI/src/Common

This library contains classes for the EcsToV0Server and V0ToEcsServer executables. These classes are the framework used to build the executables.

Classes

- DmGwConfigFile -- Read a configuration item from the EcDmV0ToEcsGateway CFG file.
- DmGwConfigItems -- This class will get items from Process Framework that have been read from the configuration file by PF.
- DmGwConvertUtil -- Static Utility methods.
- DmGwDataValue -- This class performs store and get the data value.
- DmGwDateTime -- This class performs the set time, date and get time, date.
- DmGwECSCircle -- This class performs get circle, radius, and center for the circle search.
- DmGwECSPoint -- This class performs get longitude and latitude for the point search.
- DmGwECSPolygon -- This class performs get polygon for polygon search.
- DmGwECSRectangle -- This class performs get rectangle, get north, south longitude and latitude for the rectangle search.
- DmGwEvent -- This class performs the logging events in Gateway managed server.
- DmGwExtendedSearch -- This class represents the V0 Extended Search group and consists functions to set and get the parameters for the specialized criteria group.
- DmGwGlobalGranulesOnly -- This class performs creation of object DmGwGlobalGranulesOnly.
- DmGwGroup -- Base class.
- DmGwIKSocket -- This class encapsulates the IK library routines in a class and removes all global variables required to operate.
- DmGwImageStream -- This class saves a binary image file.
- DmGwLineItem -- This class represents the V0 LINEITEM type in a Product Request message.
- DmGwManagedServer -- This server class is built as a child of the Process Framework Managed server class.
- DmGwMediaFormat -- This class performs get media format, get parameter list.
- DmGwMediaType -- This class represents an instance of an ODL MEDIA_TYPE group construct.
- DmGwODLAggregate -- This class performs get inventory search, get directory search, get browser request, get product request, get quit, get abort, get acknowledge, and extract informations from the input ODL tree.
- DmGwODLMutex -- This class provides a common mutex lock for the V0 IK/ODL code.

- DmGwODLNonAggregate -- This class performs extracting non aggregate informations from input aggregate ODL tree.
- DmGwODLParameter -- This class builds and gets parameters from ODL tree.
- DmGwODLTree -- This class builds/gets root node of an input ODL Tree.
- DmGwOrderOption
- DmGwPackage -- This class supports the following operations.
- DmGwPathRowLoc -- This class represents the V0 Search group and consists functions to set and get the parameters for the Path Row Loc group.
- DmGwPointLoc -- This class represents the V0 point location group and consists functions to set and get the latitude and longitude of the point location.
- DmGwPolygonLoc -- This class represents V0 Polygon Loc group and consists of functions to get and set the attributes using ECS types.
- DmGwProcessingOption
- DmGwRangeLoc -- This class represents V0 RangeLoc group.
- DmGwRequest
- DmGwRequestReceiver -- This class takes in a socket descriptor and request id and executes the contents of the socket in the Gateway and returns the resultant request on a new socket id.
- DmGwRSTAggregate -- This class builds all necessary ODL aggregates for the output ODL tree.
- DmGwRSTNonAggregate -- This class builds the non aggregate elements of the ODL tree for response messages.
- DmGwSpatial -- Base class for all the spatial classes that are supported by the V0 gateway.
- DmGwSpecializedCriteria
- DmGwSubsetOption -- This class represents an instance of an ODL SUBSET_OPTIONS group construct.
- DmGwTemporal -- This class constructs the temporal constraint, does conversion from V0 to ECS temporal constraint.
- DmGwTemporalVec -- This class supports the conversion of V0 temporal type which includes startday of the year and stop day of the year to multiple ECS date ranges.
- StringVec -- This class holds a lists of string pointers and forms a comma seperated list when required for SQL construction.

library CsEmMailRelA

/ecs/formal/CSS/DOF/src/EMAIL/email

This Library is used for the emails using sendmail utility and the nnpost for Bulletin board messages.

Classes

- CsBBMailRelA -- Implements CsBBMailRelA class for interactive(for operators) and development(API, for applications) interfaces to manage electronic post messages.
- CsEmMailRelA -- Implements CsEmMailRelA class for interactive(for operators) and development(API, for applications) interfaces to manage electronic mail messages.

library CsEmMailRelASh (shared)

/ecs/formal/CSS/DOF/src/EMAIL/email

This library is used for the emails using sendmail utility and the nnpost for Bulletin board messages.

Classes

- CsBBMailRelA -- Implements CsBBMailRelA class for interactive(for operators) and development(API, for applications) interfaces to manage electronic post messages.
- CsEmMailRelA -- Implements CsEmMailRelA class for interactive(for operators) and development(API, for applications) interfaces to manage electronic mail messages.

library CsFtFTPNotifySh (shared)

/ecs/formal/CSS/DOF/src/FTP/ftp_notify

This library is only used to provide information about the successful completion of FTP files from the local Host to the caller. The ftpd should be running with -lll flag. The Library uses /usr/ecs/\$MODE/CUSTOM/data/CSS directory for storing persistent files.

Classes

- CsFtFTPNotify -- This Class provides a method to extract the information regarding the successfully completed FTP files.

library CsFtFTPRelA

/ecs/formal/CSS/DOF/src/FTP/ftp

This library provides a Wrapper for FTP. User can use it for all the FTP needs, such as sending or receiving files to/from remote Hosts.

Classes

- CsFtFTPRelA -- By instantiating a CsFtFTPRelA object and sending the object messages, a program can perform programmed dialogue with a ftp process and effectively control it.

library CsFtFTPRelASh (shared)

/ecs/formal/CSS/DOF/src/FTP/ftp

This library is used for all the FTP needs, such as sending or receiving files to/from remote Hosts.

Classes

- CsFtFTPRelA -- By instantiating a CsFtFTPRelA object and sending the object messages, a program can perform programmed dialogue with a ftp process and effectively control it.

library CsFtFTPSchedObj

/ecs/formal/CSS/DOF/src/FTPSCHED

This library provides functions to use FTP at a scheduled time by the User.

Classes

- CsFtFileListObj -- This class contains all the information provided by the scheduleFileTransfer function.
- CsFtFTPSchedObj -- This class is designed to be created by the Managed Process Framework to allow developers to schedule file transfers at a given time.

library CsFtFTPSchedObjSh (shared)

/ecs/formal/CSS/DOF/src/FTPSCHED

This library provides functions to use FTP at a scheduled time by the User.

Classes

- CsFtFileListObj -- This class contains all the information provided by the scheduleFileTransfer function.
- CsFtFTPSchedObj -- This class is designed to be created by the Managed Process Framework to allow developers to schedule file transfers at a given time.

library CsSeCryptoDes

/ecs/formal/CSS/DOF/src/CRYPTO

This library contains the APIs for the DES encryption algorithm. This library is primarily used for encrypting passwords for storage. The primary functions are DesEncrypt and DesDecrypt. Both return strings containing the encrypted/decrypted text. Two other functions, DesGetMsgLen and DesGetCypherLen are provided for the application programmer to obtain the length of the returned string since it may vary slightly due to the encryption/decryption algorithm.

Classes

- CsSeCryptoDes -- This class implements both standard and modified forms of the NBS Data Encryption Standard (DES).

library CsSeCryptoDesSh (shared)

/ecs/formal/CSS/DOF/src/CRYPTO

This library contains the APIs for the DES encryption algorithm. This library is primarily used for encrypting passwords for storage. The primary functions are DesEncrypt and DesDecrypt.

Both return strings containing the encrypted/decrypted text. Two other functions, `DesGetMsgLen` and `DesGetCypherLen` are provided for the application programmer to obtain the length of the returned string since it may vary slightly due to the encryption/decryption algorithm.

Classes

- `CsSeCryptoDes` -- This class implements both standard and modified forms of the NBS Data Encryption Standard (DES).

library DmAsGwAsterReqProc

`/ecs/formal/DM/ASTGW/src/AsterReqProc`

The `lib_DmAsGwAsterReqProc` library contains classes for processing ASTER to ECS requests. These classes convert an ODL file received from the ASTER client to a `GIParameter` list. The request is then processed and the results are converted back to ODL to be sent to the ASTER client.

Classes

- `DmAsGwAsterBroRequest` -- This class processes an Aster to Ecs Browse request.
- `DmAsGwAsterBroRequestor` -- This class is used to submit a query to the SDSRV via the LIM libs.
- `DmAsGwAsterCancelRequest` -- The operations "concept" for canceling a product request is to first ask for product status, and obtain the top-level initiator request ID and then each sub-request ID.
- `DmAsGwAsterDirRequest` -- The managed server should perform a directory search request by invoking `Initialize` followed by the `SendAndReceive` function.
- `DmAsGwAsterInvRequest` -- `Submit` is called by the managed server and the output ODL tree is passed in by reference.
- `DmAsGwAsterInvRequestor` -- This class is used to submit a query to the SDSRV via the LIM libs.
- `DmAsGwAsterODLAgg` -- This class supports input ODL tree aggregates.
- `DmAsGwAsterODLNonAgg` -- This class implements `Get~()` functions that were not provided for in the base class `DmGwODLNonAggregate`.
- `DmAsGwAsterPriceEstRequest` -- This class extracts the `PRICE_ESTIMATE_REQUEST` group parameters specified in the constructor's ODL argument, converts it into a `GIParameterList` structure to submit to the LIMGR using the `DmAsGwAsterPriceEstRequestor` class. After the `GIParameterList`
- `DmAsGwAsterPriceEstRequestor` -- This is the delegatee for a given Price Estimate Request whose ODL request message was passed to its constructor.
- `DmAsGwAsterProductStatusRequest` -- The managed server should perform a product status request by invoking `Initialize` followed by the `SendAndReceive` function.

- DmAsGwAsterProductUpdate -- The operations concept for the product update is that when a previously ordered product is shipped to the client who made the original product request, a product update information message is sent and a product update acknowledge is returned as a .
- DmAsGwAsterProRequest -- This class extracts data from a Product request ODLTree message and changes it into GIParameterList format in order to submit it to the LIMGR using the DmAsGwAsterProRequestor class.
- DmAsGwAsterProRequestor -- This class is derived off of the DmAsGwAsterRequestor class, and used to submit a query to the SDSRV via the LIM libs(ASTER->ECS)).
- DmAsGwAsterRequest -- Requests that take place from ASTER to ECS.
- DmAsGwAsterRequestor -- DataManagmentGatewayAsterRequestor.
- DmAsGwAsterRSTAgg -- This class is derived from the DmGwRSTAggregate class to add build functions that were not provided in the base class.
- DmAsGwAsterRSTNonAgg -- This class is derived from the DmGwRSTNonAggregate class to add build functions that were not provided in the base class.
- DmAsGwModifyUR -- According to the ASTER GDS-ECS ICD the granule UR should be <= 50 chars.

library DmAsGwCommon

/ecs/formal/DM/ASTGW/src/Common

The DmAsGwCommon library contains utility classes and term definitions used by the Aster Gateway. Primarily, classes exist that translate between Aster and ECS terms storing commonly requested translations in cache, convert between dollars and yen, handle Aster Server event logging, and handle IK connection procedures used to transmit and receive ODL trees via sockets.

Classes

- DmAsGwAdEntry -- This class is identical to the parent DmAsGwCacheEntry class except that the destructor does a delete instead of a RecursiveDelete.
- DmAsGwAdServices -- This class aids in the translation of "ECS" IOS Advertising Service results GIParameterLists to "ASTER" GIParameterLists.
- DmAsGwCache -- This class is designed to provide a convenient mechanism for storing and retrieving CacheEntries.
- DmAsGwCacheEntry -- The DmAsGwCacheEntry class is designed to be used in conjunction with the DmAsGwCache class.
- DmAsGwErrorMsg -- This class is designed to be used in conjunction with error logging in order to reduce the number of lines of error logging code.
- DmAsGwEvent -- This class performs the logging events in Gateway managed server.
- DmAsGwTranslate -- This class is designed to translate between ECS and ASTER using the data dictionary.

- DmGwServerConnection -- This is a wrapper class that will take care of all of the IK connection procedures for transmitting and receiving odl trees via sockets.

library DmAsGwEcsReqProc

/ecs/formal/DM/ASTGW/src/EcsReqProc

The DmAsGwEcsReqProc library contains classes for all the ECS to ASTER request processing. These classes are used to send Browse, Inventory, Product, and Product Update request messages from ECS to ASTER. When the ECS request is submitted to ASTER, the constraints need to be extracted from the GIParameterList and put into an ODLTree for ASTER. After the request is processed, the results are put back into a GIParameterList and sent to ECS. The aggregate and nonaggregate classes used to create the ODLTrees are also in the DmAsGwEcsReqProc library.

Classes

- DmAsGwClientIF -- This class gets invoked by ODPRM.
- DmAsGwEcsAggregate -- This class provides operations to insert an aggregate to ODL tree.
- DmAsGwEcsBrowseRequest -- DmAsGwEcsBrowseRequestm.
- DmAsGwEcsInventoryRequest -- This class is responsible for processing an inventory request.
- DmAsGwEcsNonAggregate -- This class builds the non aggregate elements of an ODL tree.
- DmAsGwEcsProductRequest -- Declaration of class DmAsGwEcsProRequest, and all of its member functions and data members.
- DmAsGwEcsRequest
- DmAsGwInvQuery -- This class builds a SQL query using the input search parameters and submits it to the Data Dictionary database.
- DmAsGwSearchAggregate -- Also, any methods needed for preparing an Aggregate structure for submission to the Aster Server that are not included in that parent class are included here.
- DmAsGwSearchNonAggregate -- This class is publicly inherited from DmGwRSTNonAggregate and contains routines for Building parameters that are not accounted for in the parent class.
- DmGwEcsAsterRequestReceiver

library DmDdMsg

/ecs/formal/DM/DDICT/src/ddmsg

This library defines messages to be passed to the Data Dictionary.

Classes

- DmDdResultMsg -- The DDICT creates and returns a result message object.
- DmDdSchemaMsg -- Creates message object and sends it to the DDICT for schema requests for modifications to the Data Dictionary database..

- DmDdSearchMsg -- Creates message object and sends it to the DDICT for Data Dictionary search requests..
- DmDdStatusMsg -- Creates message object and sends it to the DDICT to get the status of a requests previously sent to the DDICT.
- DmDdUrRequestMsg -- Creates message object and sends it to the DDICT for Data Dictionary UR requests..
- DmDdUrResultMsg -- The DDICT creates and returns result message object in response to an UR requests.

library DmDdReqProc

/ecs/formal/DM/DDICT/src/ddreqproc

This library defines various manipulations for the data dictionary.

Classes

- DmDdArgInfoTbl -- This class is used to store/process data regarding whether or not the table has the fields collectionId, siteId, and attributeId.
- DmDdGIHelper -- This class contains methods that help in the processing of a GIParameterList.
- DmDdGroupTbl -- This class is used to store/process data for groups.
- DmDdJoinTable -- This class is used to manipulate joins.
- DmDdJPairsTbl -- This class is used to store/process data for joinable pairs table information.
- DmDdLevel1Tbl -- This class is used to store/process data for level 1 information.
- DmDdLevel2Tbl -- This class is used to store/process data for joinable pairs table information.
- DmDdLevel2TblEntry -- This class is used to store/process data for level 2 table information.
- DmDdLevel3Tbl -- This class is used to store/process data for joinable pairs table information.
- DmDdLevel3TblEntry -- This class is used to store/process data for joinable pairs table information.
- DmDdMapper -- This class performs the valid mapping for all collections in the Data Dictionary.
- DmDdParser -- This class is intended to be used in parsing the request submitted to the DDICT.
- DmDdProcMsg -- This class is used to process messages for the Data Dictionary.
- DmDdTables -- This class is used to store/process data from the tables.
- DmDdTblCols -- This class is used to store/process data for joinable pairs table information.
- DmDdValidsMapping -- This class contains the entry for each mapping.
- DmDdValidsMappingTable -- This class is used to store/process data for valid mapping.

- DmDdXrefTbl -- This class is used to store/process data for cross-reference information.
- DmDdXrefTblEntry -- This class is used to store/process data for joinable pairs table information.

library DmDdServerJit

/ecs/formal/DM/DDICT/src/ddserver

The ddserver directory contains classes that implement the server portion of the DDICT. It initializes the DDICT, connects to the Process Framework objects, and provides objects to process service requests from a client. It dispatches the received service requests and instantiates an object to process the request. When results of the request are received from the object processing the request, they are returned to the requester.

Classes

- DmDdDispatcher -- This is the customized version of the dispatcher object.
- DmDdManagedServer -- This is an example how an application can use Process Framework to create a managed server.
- DmDdRequest
- DmDdSchemaRequest
- DmDdSearchRequest -- This class receives search messages submitted to the DDICT.
- DmDdUrRequest -- This class is used by the server to process the client UR request.

library DmGwRegistryStub

/ecs/formal/DM/SIM/src/CSS

The libDmGwRegistryStub library contains definitions of methods residing in CSS. Specifically any method used by DMS servers that incur a remote procedure call. This allows us to generate test executables that link to this stub library that have IDENTICAL DMS source code, thus improving the reliability of unit testing within DMS.

Classes

- EcRgDistOptions -- This class requests the distribution media options from the EcCsRegistryServer.

library DmGwV0If

/ecs/formal/DM/GatewayCSCI/src/V0If

The DmGwV0If library contains classes for all the ECS to V0 request processing. These classes are used to send Browse, Inventory, and Product request messages from ECS to V0. When the ECS request is submitted to V0, the constraints need to be extracted from the GlParameterList and put into an ODLTree for V0. After the request is processed, the results are put back into a GlParameterList and sent to ECS. The aggregate and nonaggregate classes used to create the ODLTrees for the result messages are also in the DmGwV0If library.

Classes

- DmGwEcsAddress -- This class is responsible for storing address information.
- DmGwEcsAffiliation -- This class is responsible for storing affiliation information.
- DmGwEcsBrowseRequest -- This class is responsible for processing a V0 Browse request.
- DmGwEcsInvRequest -- This class is responsible for processing a V0 Inventory request.
- DmGwEcsLineItem -- This class is responsible for storing line item information.
- DmGwEcsODLAggregate -- This class is responsible for building a sub-tree in ODL tree.
- DmGwEcsODLNonAggregate -- This class is responsible for inserting attribute values into ODL tree.
- DmGwEcsProductRequest -- This class is responsible for processing a V0 Product request.
- DmGwEcsResultAggregate -- This class is responsible for extracting sub-tree from ODL tree.
- DmGwEcsResultNonAggregate -- These functions are responsible for extracting values from ODL tree.
- DmGwEcsServRequest -- This class is a base class from which specialized versions are derived.

library DmGwV0Util

/ecs/formal/DM/GatewayCSCI/src/V0Util

The DmGwV0Util library contains the implementation of functions used by the ECS to V0 Gateway for client/server interfaces.

library DmLmAsterIfFactory

/ecs/formal/DM/LIMGR/src/factory

The DmLmAsterIfFactory library contains classes that create the correct type of external interface used by the LIMGR to access external interfaces.

Classes

- DmLmInterfaceFactory -- This class creates an external interface object of a certain type.

library DmLmDbi

/ecs/formal/DM/LIMGR/src/dbi

The DmLmDbi library contains utilities for interfacing and connecting with the DDICT database. Queries can either be submitted as a GIPparameter or a string.

Classes

- DmLmDbiBase -- This class contains utilities for interfacing with the Data Dictionary.
- DmLmDbiUtils -- This class contains utilities for interfacing with the Data Dictionary.

- DmLmIntQuery -- This class provides the functionality to Submit a query in string format to the data dictionary, then retrieve the results after submitting.

library DmLmExtIf

/ecs/formal/DM/LIMGR/src/extIf

The DmLmExtIf library contains classes that allow the LIMGR to interface with external information managers. Specifically, the information managers are the ASTER IMS, Science Data Server, Document Server, Gateway, and V0 Gateway.

Classes

- DmLmAsterServIF -- This class allows the LIMGR external interface library to connect to the Aster IMS.
- DmLmDataServerIF -- This class object is a specialization of the DmLmExtIFBase class.
- DmLmExtIFBase -- This class object is the base class from which specialized versions (e.g. DmLmV0ServIF) are derived.
- DmLmGranuleUR -- The DmLmGranuleUR class represents a granule which is the result from inventory search.
- DmLmGranuleUrProvider -- Definition of the class DmLmGranuleUrProvider.
- DmLmInfoMgr -- This class represents the server for DMS.
- DmLmInfoMgrIF -- quest object representing this request.
- DmLmV0ServIF -- This class allows the ECS-V0 gateway to interface with the V0 data server.

library DmLmGlobal

/ecs/formal/DM/LIMGR/src/global

The global CSC provides classes for DM subsystem to handle the global variables. This is the both client and server sides library.

Classes

- DmLmCollection -- This class is an entry for the collection table.
- DmLmCollectionTable -- The class contains a vector which is called collection table.
- DmLmConfiguration -- Through the object, configuration of the information manager can be inquired, such as its name, type, UR, PF configuration file and etc.

library DmLmGlobalSh (shared)

/ecs/formal/DM/LIMGR/src/global

The global CSC provides classes for DM subsystem to handle the global variables. This is the both client and server sides library.

Classes

- DmLmCollection -- This class is an entry for the collection table.
- DmLmCollectionTable -- The class contains a vector which is called collection table.
- DmLmConfiguration -- Through the object, configuration of the information manager can be inquired, such as its name, type, UR, PF configuration file and etc.

library DmLmInfoMgrFactory

/ecs/formal/DM/LIMGR/src/factory

The DmLmInfoMgrFactory library contains classes that create the correct type of external interface used by the LIMGR to access external interfaces.

Classes

- DmLmInterfaceFactory -- This class creates an external interface object of a certain type.

library DmLmMapper

/ecs/formal/DM/LIMGR/src/mapper

The library contain a class used to map client attribute names to science data server attribute names and map ECS attribute names and values to VO attribute names and values.

Classes

- DmLmMapper -- The class provides methods to map attribute names and values from ECS to VO or ASTER term, or vice versa.

library DmLmMsg

/ecs/formal/DM/LIMGR/src/msg

The msg CSC provides classes for DM subsystem server and client. Client uses them to create the request objects sending to the server, and server uses them to create the result object replying to the client. This library is used both sides of client and server.

Classes

- DmLmAcquireMsg -- Creates message object and sends it to the SDSRV for acquire requests.
- DmLmBrowseMsg -- Creates message object and sends it to the SDSRV for browse requests.
- DmLmExitMsg -- Creates message object and sends it to the SDSRV for exit requests.
- DmLmInspectMsg -- Creates message object and sends it to the SDSRV for inspect requests.
- DmLmNextBrowseMsg -- Creates message object and sends it to the SDSRV for next browse requests.
- DmLmResultMsg -- The SDSRV creates and returns result message object inresponse to acquire, browse, exit, inspect, next browse, search, and status requests..
- DmLmSearchMsg -- Creates message object and sends it to the SDSRV for search requests.

- DmLmStatusMsg -- Creates message object and sends it to the SDSRV for status requests.

library DmLmReqProc

/ecs/formal/DM/LIMGR/src/reqproc

The libDmLmReqProc library contains classes which handle the routing of local DMS requests. Any message received is identified and examined to determine the local destination of that request within DMS and then sent there.

Classes

- DmLmAcquireResultSet -- This class provides methods to set or get information of results after a search request.
- DmLmAttributeCollectionsType
- DmLmBrowsePlan -- This class provides methods to create request and integrate plans for the browse request.
- DmLmBrowseResultSet -- This class provides methods to set or get information of results after a search request.
- DmLmInspectPlan -- This class provides methods to create request and integrate plans for the inspect request.
- DmLmParsedReqInfo
- DmLmParsedReqInfoType -- Information relevant to a parsed SEARCH/GUIDE request is stored in this class.
- DmLmParser -- The DmLmParser is a class for parsing and validating the request list.
- DmLmPlanner -- This class provides methods to create, request, and integrate plans for a search or product request.
- DmLmProcessor -- The public operations include to process the submission of the request to different target servers, create the result data store and integrate the results.
- DmLmProcessPlan -- This class is an entry of request processing.
- DmLmProductPlan -- This class provides methods to create request and integrate plans for the product request.
- DmLmResultSet -- This class provides methods to set or get information of results after a search request.
- DmLmSearchPlan -- This class provides methods to create request and integrate plans for the search request.
- DmLmSearchResultSet -- This class provides methods to set or get information of results after a search request.
- DmLmSubmittedRequest -- The class provides methods for request processing to submit a request to different types of target server.

library DmLmV0IfFactory

/ecs/formal/DM/LIMGR/src/factory

The DmLmV0IfFactory library contains classes that create the correct type of external interface used by the LIMGR to access external interfaces.

Classes

- DmLmInterfaceFactory -- This class creates an external interface object of a certain type.

library DpAt

/ecs/formal/PDPS/DPS/SSIT/src/LIB

This DpAt library holds the SSIT classes that are called by various SSIT tools and programs.

The DpAt library contains classes and methods that provide UNIX system call capabilities, GUI exception/error handling and Process Framework initialization/environmental setup.

Classes

- DpAtDirectoryInfo -- This class provides access to the UNIX directory structure.
- DpAtMgrMessageDialog -- This defines a Message Dialog box for the SSIT Manager and other SSIT GUIs.
- UIComponent -- Base class for all C++/Motif UI components.

library DpAtDsrv

/ecs/formal/PDPS/DPS/SSIT/src/Dsrv

This library holds the object through which almost all SSIT calls to interface with the Data Server.

Classes

- DpAtDsrv -- This class's methods are the interface between the SSIT environment and the Data Server..

library DpAtMetadata

/ecs/formal/PDPS/DPS/SSIT/src/Metadata

This library holds the objects (DpAtDatabase and DpAtOperationalMd) through which most SSIT PDPS database updates are performed. It also contains the object (DpAtScienceMd) which parses and validates the ODL that defines PGEs to PDPS.

The DpAtDatabase class is called by the ODL parsing programs to insert new PGE information and update existing PGEs in the PDPS database. DpAtOperationalMd provides checking and updating of the data from the Operational Metadata GUI. DpAtScienceMd provides ODL parsing and validation.

library DpAtSSAP

/ecs/formal/PDPS/DPS/SSIT/src/SsapGui

This library contains the objects needed for the SSAP Editor application.

Classes

- DpAtAccess -- This class provides access control and access logs for the current user of the SSAP GUI.
- DpAtBatchFile -- This file contains the attributes and function prototypes of all operations of the DpAtBatchFile class.
- DpAtEditSSAPFileList -- This class provides the functionality for adding & deleting files from the specified SSAP.
- DpAtSSAPACFORM -- This class defines the Associated Collections window for the SSIT SSAP Editor GUI.
- DpAtSSAPBatchModeFILE
- DpAtSSAPCreateFORM -- This class defines the Create Window for the SSIT SSAP Editor Gui.
- DpAtSSAPCreateWithFORM -- This class defines the Create With window for the SSIT SSAP Editor GUI.
- DpAtSSAPDateHelper -- This class provides the operations needed to convert from one date string to another.
- DpAtSSAPFrame -- This class defines most of the SSIT SSAP Editor GUI.
- DpAtSSAPManager -- This class provides the primary access from SSIT to the Science Software Archive Packages stored at the Data Server.
- DpAtSSAPMetadata -- This class provides access from SSIT to the Science Software Archive Package metadata.
- DpAtSSAPUIComponent -- Base class for all C++/Motif UI components.

library DpDeletionClient

/ecs/formal/PDPS/DPS/PRONG/src/DataMgmt/DeletionPF

The library containing the client side operations to issue requests to the deletion server.

This library contains two client usable APIs: one to issue a delete request on short and long delete times and another to delete a list of granules from the PDPS database. This first API is used by EcDpPrDM in the destage job, to issue deletes for interim granules. The second API is used by Production Request Editor, when deleting DPRs (and their associated granules).

Classes

- Deletion_1_0 -- This is a class that is used for communications with other servers.
- DpDeletionProxy -- Converts the data types into the required OODCE formats before calling the underlying OODCE client-side function and converts the return DCE data back to what's required.

library DpPrDbPf

/ecs/formal/PDPS/DPS/PRONG/src/Common

This library contains the PDPS database interface classes for use with the Process Framework access to the database.

Classes

- DpPrDbColVal -- This class is used to create a column-value structure so that the database interface class can use to map the object attribute to appropriate column in the table.
- DpPrDbColValList -- This class contains a list of DpPrDbColVal objects.
- DpPrDbConnectRecord -- This class is used to support searching and retrieval of information from the database.
- DpPrDbIF -- Must undefine external definition of Status before defining it below.
- DpPrDbMaster -- This base class monitors all the connections to the database.

Templates

- DpPrDbInterface -- This class provides necessary methods to manipulate database table from C++ application program.

library DpPrDM

/ecs/formal/PDPS/DPS/PRONG/src/DataMgmt

This library supplies utility operations to manage data within DPS.

These operations are used for the two threads within the administrative job, EcDpPrDM: those of staging and destaging data, and managing their disc resources via calls to resource manager. There are no static operations, so an object must be instantiated to take advantage of the public operations available.

Classes

- DpPrDataGransGroup -- This class defines attributes and operations for a group of data granules based on groupId and DataType name.
- DpPrDataManager -- This class defines attributes and operations for initializing and managing data granules required by a PGE during its execution.
- DpPrDMPfClient -- This class invokes stage and destage jobs in the AUTOSYS job box.
- DpPrFile -- Since this class is a persistent class, it defines attributes and operations for manipulating the instances of objects as the entries in the Sybase DataBase.
- DpPrGranuleLocator -- This class provides an interface to the DpPrGranuleLocation table and some convenience methods for determining whether or not data is local (in the DAAC and visible), nearby (in the DAAC but not visible), or need to be staged from the DSS.

library DpPrDssIF

/ecs/formal/PDPS/DPS/PRONG/src/DssIF

This library contains the helper object for PDPS to interact with science data server.

Clients use this object, rather than making a direct call, so that the interface can be encapsulated within a single class for maintainability. This library includes calls to APIs from both DSS and IOS.

Classes

- DpPrDSSInterface -- This class provides a central (NON-PF-CLIENT) class for all interactions with either DSS or the DSS simulator.
- DpPrRpcID -- This object encapsulates the EcUtRpcID for PDPS.

library DpPrEM

/ecs/formal/PDPS/DPS/PRONG/src/ExecMgmt

This library supplies utility operations to manage PGE execution within PRONG.

This library is mostly used by DpPrEM.cxx, which creates the program EcDpPrEM. Using this library, through the instantiation of and using its public API, allocation, deallocation, pre and post processing can be accomplished when running a PGE.

Additionally, using the class, the current state of a particular dpr, whether it has finished allocating resources or whether it has post processed the pge's outputs etc., can be obtained.

Also, several of class's methods are static methods. These methods can be used to execute a system command, such as to obtain unix file information.

Classes

- DpPrExecutable -- This class is used to maintain the state of the science software components which are crucial to support the proper runtime operation of a PGE.
- DpPrExecutionManager -- This DpPrExecutionManager class is the interface class for other classes which require execution services.
- DpPrLock -- This class provides general purpose semaphore for critical sections of code.
- DpPrPcf -- This class is used to maintain the state of the Process Control File (PCF), which provides critical runtime information to the PGE, while the science software is executing.
- DpPrPcfGran -- DpPrPcfGran represents a granule (inputs & outputs).
- DpPrPge -- This class is used to maintain the state of the completed PGE and as such provides for the insertion and deletion of the science software for the local machine and controls execution of the PGE on that platform.
- DpPrUtil -- This class provides utility methods used by EM.
- PlCompletion -- This class is used to maintain the termination results of a DPR run.

library DpPrException

/ecs/formal/PDPS/DPS/PRONG/src/Exception

This library provides APIs to catch and handle signals.

Exception is a misleading name. This library does not catch C++ exceptions. It catches unix signals (mainly SIGHUP) and programs which use this library can perform any cleanup before exiting.

Classes

- DpPrInterrupt -- This class manages PRONG administrative job's process interrupt state. It manipulates the process's interrupt signal handling characteristics to provide a transparent mechanism for a running process to perform a graceful shutdown when requested.

library DpPrJM

/ecs/formal/PDPS/DPS/PRONG/src/JobMgmt

The Job Management library provides service routines for the Job Management Server to manage the flow of jobs in all AutoSys instances for a given mode. These services include creating jobs, canceling jobs, releasing queued jobs, getting the status of jobs and controlling job execution through AutoSys. These routines are not called directly by clients, instead, a proxy library is provided with similar interfaces to the methods defined in this library.

Classes

- DpPrAutosysMap -- This class assigns and retrieves the maps between resource strings and the IDs of AutoSyses managing jobs running on them.
- DpPrAutosysMapList -- This persistent class stores and provides access methods to the list of AutoSys ID & string name mappings.
- DpPrCotsManager -- This class provides an interface to the AutoSys COTS product to add, delete, and modify jobs and machines; control job processing; retrieve job definition and processing status; generate reports; and send alarms.
- DpPrCreateQObj -- This class contains information necessary for constructing AutoSys jobs.
- DpPrCreationQueue -- This class manages queues of CreateDprJob parameters waiting to be used to create jobs in AutoSys.
- DpPrCreationQueueByAutosys -- This is a priority queue (sorted vector) of CreateDprJob parameter lists.
- DpPrDbTableIf -- This class defines the interface to Sybase for the DpPrDbInterface class.
- DpPrJIL -- AutoSys JIL interface.
- DpPrMaxJobs -- This is the persistent class that defines the attributes and operations for manipulating the instances of objects that are stored in the DpPrMaxJobs database table.
- DpPrScheduler -- This class provides PLANG Planning Workbench and Subscription Manager components a programmatic interface to the PRONG.

library DpPrQaMonitor

/ecs/formal/PDPS/DPS/PRONG/src/QaMonitor

This library holds the functions and attributes that all the QA Monitor GUI to perform its various functions.

The actual GUI code is in a separate section, the classes within this library provide the function to the QA Monitor GUI.

Classes

- DpPrQaDataGranule -- It defines attributes and operations of data parameters returned from DataServer when operator tries to query data for a particular data type during a period of starting and ending day.
- DpPrQAGranuleQaFlags -- These methods and attributes support display and update of granule parameters.
- DpPrQaMonitor -- The DpPrQaMonitor is what the Quality Assurance (QA) position uses to query all data granules for a particular data type within a date interval, retrieve data granules from Data Server to local disk, and to view data products created with QA metadata

library DpPrRM

/ecs/formal/PDPS/DPS/PRONG/src/RsrcMgmt

This library supplies utility operations to manage computer resources for PDPS.

There is only ever one instance of a Resource Manager active at any one time. A resource lock prevents other operations from creating multiple Resource Manager objects.

Classes

- DpPrComputer -- This class is used to represent the set of computer hardware that is being used for science software processing within the Processing System.
- DpPrCpuRamAllocation -- This class is used to maintain the individual records of CPU and RAM usage, which are used to maintain the integrity of processor and memory allocations and deallocations.
- DpPrDiskAllocation -- This class is used to maintain the individual records of disk storage usage which are used to maintain the integrity of storage allocations and deallocations.
- DpPrDiskPartition -- This class is used to represent the set of disk storage devices that are being used to contain the files needed to run a collective set of software known as a PGE.
- DpPrResource -- This base class is used to capture the similar features of the derived classes, and to provide for future expansion.
- DpPrResourceLock -- This class is used to implement locking of the Resource Management database tables on a per-job basis.
- DpPrResourceManager -- This interface class provides an abstract set of operations for effectively managing the collection of processing and storage resources.
- DpPrString -- Abstract representation of one or more actual machines.

library DpPrTools

/ecs/formal/PDPS/DPS/PRONG/src/Tools

This library contains global utility functions.

This library contains global utility functions to get directory listing, space available on a given device, ability to set the application log file name etc.,

library DpPrUpdateStatus

/ecs/formal/PDPS/DPS/PRONG/src/Common

This library provides functionality to update the status of a Request maintained in the MSS database.

Classes

- DpPrUpdateStatus -- This class provides functionality to update the status of a Request maintained in the MSS database.

library DsAdSh (shared)

/ecs/formal/DSS/sdsrv/src/ad

This library provides an administrative/M&O interface to science data server requests. It also provides a collection mechanism for retrieving and manipulating multiple descriptors. It is mainly used by EcDsSdSrvGui for installing ESDTs and monitoring the Science Data Server requests.

Classes

- DsAdDataTypeCollector -- This class provides a collection mechanism for retrieving and manipulating multiple Earth Science Data Type descriptors.
- DsAdSciRequest -- Provides an interface to access Science Data Server Request information like request ids, priority, domain and state request information as well as command service, command category and number of commands.
- DsAdSciRequestInterface -- This class provides an administrative/M&O interface to science data server requests.

library DsBtSh (shared)

/ecs/formal/DSS/sdsrv/src/bt

This library implements a thread which notifies the subscription server of events. Notification requests are inserted in the queue by the processing threads in the data server and are removed and processed by this thread in the background. This processing is done in a separate thread so that if the subscription server is not responding, the processing of the request can proceed, and the sending of the event can be retried ad infinitum until the subscription server comes back. In addition, the queue is mirrored in the databawse, so that if the dataserver must be restarted before the events are sent, the queue will not be lost.

Classes

- DsBtEventCleanUpJob -- This is the DsBtEventCleanUpJob header file.
- DsBtJob -- This is the DsBtJob header file.
- DsBtJobManager -- This is the DsBtJobManager header file.
- DsBtRequestCleanUpJob -- This is the DsBtRequestCleanUpJob header file.
- DsBtSbsrvNotifier -- This class (actually a collection of static functions and variables) implements a thread which notifies the subscription server of events.

library DsCI

/ecs/formal/DSS/sdsrv/src/cl

This library contains classes which make up the Science Data Server's (SDSRV) public interface. These classes are used by clients to connect, build commands, and submit requests to the SDSRV.

Classes

- DsCIAcquireCommand -- Provides an encapsulated mechanism for transporting ESDT acquire command parameters through the SDSRV client interface.
- DsCICommand -- Provides a client view of the shared command implementation class, which is transferred to the server upon submission.
- DsCIDescriptor -- Client side class that provides an interface to the descriptor.
- DsCIESDTAddRequest -- This class is used to add an ESDT Reference to the ESDTReferenceCollector as well as adding an ESDT to the working collection.
- DsCIESDTReference -- This object is a reference to an ESDT that is within a Data Server's holdings. This object provides services that are homogeneous for all ESDTs..
- DsCIESDTReferenceCollector -- This class contains the specialized functions that pertain to management of state (the working collection on the server side) by mimicing that state on the client machine.
- DsCIESDTReferenceCollectorVector -- This object is a vector that contains a set of ESDT Reference Collectors.
- DsCIESDTReferenceVector -- Holds ESDT type-level information and pointers to ESDTs.
- DsCIESDTRemoveRequest -- This class is used to remove an ESDT Reference from the working collection.
- DsCIGenConnector -- This class encapsulates communications with a data server.
- DsCIGenRequestInt -- The definition of the DsCIGenRequestInt class, which represents the generic portion of the client side's interface to request objects..
- DsCIGenSuspendor -- Provides specialized collector types requiring state management.
- DsCIInsertCommand -- Provides an encapsulated mechanism for transporting ESDT insert command parameters through the SDSRV client interface.
- DsCIQuery -- Provides a convenient way for clients to make search requests.

- DsClRequest -- It is composed of a sequence of DsClCommand objects (created by the client), which are the basic execution units.
- DsClRestartNotifier -- This class is used by the client to notify the Science Data Server of its start up temperature.
- DsClSubsetCommand -- Provides an encapsulated mechanism for transporting ESDT subsetting command parameters through the SDSRV client interface.
- DsClTypeInfo -- Container class for product-type-level metadata.

library DsClSh (shared)

/ecs/formal/DSS/sdsrv/src/cl

This library contains classes which make up the Science Data Server's (SDSRV) public interface. These classes are used by clients to connect, build commands, and submit requests to the SDSRV.

Classes

- DsClAcquireCommand -- Provides an encapsulated mechanism for transporting ESDT acquire command parameters through the SDSRV client interface.
- DsClCommand -- Provides a client view of the shared command implementation class, which is transferred to the server upon submission.
- DsClDescriptor -- Client side class that provides an interface to the descriptor.
- DsClESDTAddRequest -- This class is used to add an ESDT Reference to the ESDTReferenceCollector as well as adding an ESDT to the working collection.
- DsClESDTReference -- This object is a reference to an ESDT that is within a Data Server's holdings. This object provides services that are homogeneous for all ESDTs..
- DsClESDTReferenceCollector -- This class contains the specialized functions that pertain to management of state (the working collection on the server side) by mimicing that state on the client machine.
- DsClESDTReferenceCollectorVector -- This object is a vector that contains a set of ESDT Reference Collectors.
- DsClESDTReferenceVector -- Holds ESDT type-level information and pointers to ESDTs.
- DsClESDTRemoveRequest -- This class is used to remove an ESDT Reference from the working collection.
- DsClGenConnector -- This class encapsulates communications with a data server.
- DsClGenRequestInt -- The definition of the DsClGenRequestInt class, which represents the generic portion of the client side's interface to request objects..
- DsClGenSuspendor -- Provides specialized collector types requiring state management.
- DsClInsertCommand -- Provides an encapsulated mechanism for transporting ESDT insert command parameters through the SDSRV client interface.
- DsClQuery -- Provides a convenient way for clients to make search requests.

- DsClRequest -- It is composed of a sequence of DsClCommand objects (created by the client), which are the basic execution units.
- DsClRestartNotifier -- This class is used by the client to notify the Science Data Server of its start up temperature.
- DsClSubsetCommand -- Provides an encapsulated mechanism for transporting ESDT subsetting command parameters through the SDSRV client interface.
- DsClTypeInfo -- Container class for product-type-level metadata.

library DsCn

/ecs/formal/DSS/sdsrv/src/cn

The library was used to generate files which install ESDT configuration information for each ESDT previously. Presently the database is used instead to store the configurations. Therefore the library will not be used by SDSRV, but it is still used by STMGT.

Classes

- DsCnConfiguration -- Used to access a server specific configuration file.

library DsCnSh (shared)

/ecs/formal/DSS/sdsrv/src/cn

This library was used to generate files which install ESDT configuration information for each ESDT previously. Presently the database is used instead to store the configurations. Therefore the library will not be used by SDSRV, but it is still used by STMGT.

Classes

- DsCnConfiguration -- Used to access a server specific configuration file.

library DsCsSh (shared)

/ecs/formal/DSS/sdsrv/src/cs

This library is used by the HDFEOS server which supplies subsetting services to DLL or other clients. Access to the subsetting is via Cs client calls to the server pool managed by the Science Data Server. This library contains the low level HDFEOS server code responsible for Conformant and NonConformant subsetting, metadata update for subsetting granule, and HDFEOS reformatting. The DsCs library is dependent on Hc, Hr and l7_dcs libraries.

Classes

- DsCsAttribute -- This class holds all the attribute information of a HDFEOS object.
- DsCsConformant -- This class implements operations for the Conformant interface.
- DsCsConformantImp -- This class implements operations for the Conformant file interface.
- DsCsCSDT -- This class provides common interface methods for the DLL to interact with the original file to be subsetting and the subsetting file.

- DsCsCSDTImp -- This class is an abstract superclass of the DsCsHDFEOSImp and DsCsNonHDFEOSImp classes.
- DsCsDimMap -- This class fetches the mappings from the HDFEOS input file and can define them in the output HDFEOS file.
- DsCsDimMgr -- This is an abstract class which is the superclass of the classes DsCsSwathMgr and DsCsGridMgr.
- DsCsDimSum -- This is a summary of all dimension information.
- DsCsField -- This class stores all information related to a HDFEOS field.
- DsCsGridMgr -- This class is responsible for creating grids resulting from subsetting, including creating new grid names, dimension names, field names and dimension map names etc..
- DsCsHDFEOSImp -- This class provides a placeholder for the grid, swath and point data objects in HDFEOS files.
- DsCsHdfEosSDS
- DsCsHDFEOSUtil -- This class is for utilities common to HDFEOS users.
- DsCsIdxMap -- This class fetches the indexed mappings from HDFEOS input file and defines them in the output HDFEOS file.
- DsCsMask -- All masks which are available for inter-DAAC compression are stored as members of this class.
- DsCsMetadataUpdate -- This class implements updating the inventory metadata in the HDFEOS output file after subsetting calls made to the HDFEOS server, as well as copying necessary attributes (e.g. global attributes) and their values to the subsetted output file.
- DsCsMgr -- This class provides the framework for the subclasses to perform various subsetting functions, to create output objects to accommodate the subsetting results, to write the new objects into files etc..
- DsCsNonConformant -- This interface is for products which do not use the naming conventions in the subsetting white papers.
- DsCsNonConformantImp -- This class implements operations for the NonConformant file subsetting interface.
- DsCsNonHDFEOSImp -- This class provides a placeholder for non-HEDEOS data objects in the file.
- DsCsOutputDimSDS -- This class provides the required interfaces with Grid and Swath objects in HDFEOS as well as SDS in HDF output files.
- DsCsOutputFile -- This class is used by the DsCsHDFEOSImp class and provides operations that interface with staging disk classes for storage management.
- DsCsOutputGrid -- This class provides all required interfaces (defining, creating, writing etc.) with grid data objects in HDFEOS output files.
- DsCsOutputPoint -- This class provides all required interfaces (defining, creating, writing etc.) with point data objects in HDFEOS output files.
- DsCsOutputSDS

- DsCsOutputSwath -- This class provides all required interfaces (defining, creating, writing etc.) with swath data objects in HDFEOS output files.
- DsCsPointMgr -- This class is responsible for creating the output points based on the constraint parameters.
- DsCsProjection -- This class is used to extract, store and define grid projection information like projection type, zone code, spheroid type, projection parameters, pixel location and origin of data.
- DsCsPtLevel -- DsCsPtLevel class is responsible for storing all the level information for a level in the output point.
- DsCsSrcDimSDS -- This is a superclass with subclasses DsCsSrcGrid and DsCsSrcSwath.
- DsCsSrcGrid -- This class provides all required interfaces (inquiring, reading etc.) with grid data objects in HDFEOS source file.
- DsCsSrcPoint -- This class provides all required interfaces (inquiring, reading, extracting etc.) with grid data objects in HDFEOS source file.
- DsCsSrcSDS -- This is a superclass with subclasses DsCsSrcDimSDS and DsCsSrcPoint.
- DsCsSrcSwath -- This class provides all required interfaces (inquiry, reading, extracting etc.) with swath data objects in HDFEOS source files.
- DsCsSwathMgr -- This class is responsible for creating swaths resulting from subsetting, including creating new swath names, dimension names, field names and dimension map names etc..
- ESĐTAnyProduct -- This is the public interface used by the Client to provide subsetting functionality.

library DsDbSh (shared)

/ecs/formal/DSS/sdsrv/src/db

This library encapsulates the Sybase Open Client calls from the rest of the SDSRV. All SDSRV database access is made through this library (both dynamic SQL and Stored Procedure calls).

Classes

- DsDbAttributeTableXref -- Attribute table entry class.
- DsDbAttributeToTableVector -- Attribute to table hash table class.
- DsDbInterface -- Database (Sybase) interface class to encapsulate database related services such as: connect, execute, fetch result.

library DsDcSh (shared)

/ecs/formal/DSS/sdsrv/src/l7_dcs

This library is used by subsetting code which make reformatting calls to convert HDF-EOS file format to HDF file format for subsetted HDF files. At present, it is only for Landsat7 data conversion.

library DsDdBSh (shared)

/ecs/formal/DSS/distribution/src/base

This library contains base objects used in both the client and server components of DDIST.

Classes

- DsDdDbConnectionPool -- DDIST base class for the database connection pool.
- DsDdDbInterface -- Database Interface class.
- DsDdDistFileB -- Holds the client side file-level information for the distribution request.
- DsDdDistListB -- Implements the client side of the distribution request, holding the list of granules for the request.
- DsDdDistRequestB -- DDIST base class for the client-server requests; holds information relating to the request.
- DsDdGranuleB -- Holds the client side granule-level information and the list of files which comprise the granule; files can be from archives or working areas under STMGT control.
- DsDdRequestListB -- The DDIST base class for the client-server request list.

library DsDdCSh (shared)

/ecs/formal/DSS/distribution/src/client

This is the client side library.

Classes

- DsDdDistFileC -- Holds the client side file-level information for the distribution request.
- DsDdDistListC -- Implements the client side of the distribution request, holding the list of granules for the request.
- DsDdDistRequestC -- DDIST client side of request client-server interface; holds information relating to the request.
- DsDdEventLogC -- Implements the client side of the Event Logging error/messages in to the data distribution database.
- DsDdEventLogListC -- Implements the client side of the Event Log's client list..
- DsDdGranuleC -- Holds the client side granule-level information and the list of files which comprise the granule; files can be from archives or working areas under STMGT control.
- DsDdRequestListC -- DDIST class for the client request list.
- DsDdRequestMgrBaseC -- Client-side request list for DDIST client-server interface.
- DsDdRequestMgrC -- Client-side request list subclass for DDIST client-server interface.
- DsDdRequestMgrCFhExecutor
- DsDdRequestMgrCProxy -- This class provides a DCE-independent interface to the widget on the client-side.
- DsDdTapeIdC -- Implements the client side of the Tape ID's associated with a distribution request..

- DsDdTapeIdListC -- Implements the client side of the Tape ID's client list associated with a distribution request..
- DsDdTapeRequestListC -- DDIST class for the client 4mm or 8mm request list.
- DsDdTestSuite -- The DsDdTestSuite is a collection of test methods collected into a class for ease of maintenance and testing.

library DsDdISh (shared)

/ecs/formal/DSS/distribution/src/idl

Classes

- DsDdRequestMgrIDL_2_0
- DsDdRequestMgrIDL_2_0_ABS
- DsDdRequestMgrIDL_2_0_Mgr

library DsDdSSh (shared)

/ecs/formal/DSS/distribution/src/server/main

The main directory holds the source code for most of the server program. The sibling directories contain specialized code for the different media types.

Classes

- DsDdActiveQueue -- This class is the class for the queue of active data distribution requests.
- DsDdAuxState -- Class designed to support auxiliary state info with persistence for recovery and for request resumption.
- DsDdBaseQueue -- This class is the base class for queues.
- DsDdCBCache -- This class stores the capacity and block size for each media type.
- DsDdConfiguration -- The DsDdConfiguration class is a singleton class which manages configuration information for the DDIST subsystem.
- DsDdDCERequestMgrConcrete -- This class provides (server-side) definitions for the distributed functions for a widget.
- DsDdDistFileS -- Holds the client side file-level information for the distribution request.
- DsDdDistListS -- Implements the server side of the distribution request, holding the list of granules for the request and estimation information.
- DsDdDistRequestS -- DDIST server side of request client-server interface; holds information relating to the request.
- DsDdDistStatus -- The DsDdDistStatus class defines the status information associated with a distribution request.
- DsDdDoneQueue -- This class is the class for the queue of shipped data distribution requests.
- DsDdGranuleS -- Holds the server-side granule-level information and the list of files which comprise the granule; files can be from archives or working areas under STMGT control.

- DsDdHoldQueue -- This class is the class for the queue of holding data distribution requests.
- DsDdLog -- Dont forget to inline.
- DsDdMedia -- Base class for defining media-specific processing associated with a particular media type.
- DsDdMediaBuilder -- Handles loading the shared object executable file for media-specific processing and creating an object instance of the specific class.
- DsDdMediaDist -- DDIST packaging class which holds a list of granules which will be distributed on the media.
- DsDdMediaFactory -- The DsDdMediaFactory class is a singleton class which handles the creation of DsDdMedia-based classes by loading and managing entry points to DLLs which hold each of the DsDdMedia-based class code, structure, constructors, destructors, methods, etc.
- DsDdPfConfigFile -- The DsDdPfConfigFile class is a singleton class.
- DsDdPriorityQueue -- This class is the class for the queue of pending data distribution requests.
- DsDdPriorityThread -- This class is the class for the object that allows requests to be throttled by priority level.
- DsDdReadyToShipQueue -- This class is the class for the queue of data distribution requests which are ready to ship.
- DsDdRequestListS -- DDIST class for the server request list.
- DsDdRequestMgrReal -- This class represents the server-side DCE-independent widget.
- DsDdRequestMgrServer -- Real was already being brought in through Concrete.
- DsDdRequestStatus -- Class for defining the status of a request from the view of DDist down.
- DsDdScheduler -- This class is responsible for multi-threading requests.
- DsDdSetLogFname
- DsDdTestSuiteS -- The DsDdTestSuite is a collection of test methods collected into a class for ease of maintenance and testing.

library DsDe1Sh (shared)

/ecs/formal/DSS/sdsrv/src/de1

This internal library contains the method to do validation, installing ESDT, sending metadata to IOS,DMS and SBSRV and generate MCF to ingest.

Classes

- DsDeCustomizer -- Base class that provides the ability to customize features of an ESDT.
- DsDeCustomizerVector -- This is a container class for customizers for a given datatype.
- DsDeDataDictController -- This class encapsulates SDSRV's interaction with DDICT. Policy decisions are placed here to unburden the Descriptor classes..

- DsDeDDCustomizer -- This class provides the ability to customize the data dictionary information for an ESDT.
- DsDeDescriptor -- This is the server side class that provides an interface to the descriptor.
- DsDeDescriptorBuilder -- This class reads in the ASCII file that defines the descriptor for a particular type.
- DsDeDescriptorImp -- This class contains all of the attributes and associated implementation of the descriptor class.
- DsDeDescriptorModifier -- This class provides the ability to contain the modified information of descriptor for an ESDT.
- DsDeDescriptorSet -- DsDeDescriptorSet is the keeper of all descriptors.
- DsDeEventCustomizer -- This class provides the ability to configure the an event recognized by the ESDT.
- DsDeEventCustomizerVector -- This is a container class for the set of event customizers for a given datatype.
- DsDeEventTable -- This class holds the cross reference of event names to event IDs.
- DsDeExpressionRule -- Contains the functions common to all expression rules including real and symbolic.
- DsDeIOSController -- Encapsulates the Science Data Server's interface to the Advertising Server.
- DsDeMetadataCustomizer -- This class provides the ability to customize metadata attributes for an ESDT.
- DsDeMetadataCustomizerVector -- This is a container class for the set of metadata customizers for a given datatype.
- DsDeRealRuleFactory -- This class is a specialization of the RuleFactory class.
- DsDeRule -- This class provides the interface by which all rules are created.
- DsDeRuleFactory -- This class provides the interface by which all rules are created.
- DsDeServiceCustomizer -- Provides the ability to customize the services that an ESDT recognizes.
- DsDeServiceCustomizerVector -- This class provides the ability to configure the services that an ESDT provides.
- DsDeStructCustomizer -- This class provides the ability to customize the CSDT type and format that an ESDT is implemented with.
- DsDeStructCustomizerVector -- Provides the ability to configure the services that an ESDT provides.
- DsDeSymbolicExpressionRule -- Specifies an expression for value validation.
- DsDeSymbolicMatchRule -- Specifies a set of valid values for an attribute.
- DsDeSymbolicRangeRule -- Specifies a valid range which includes a minimum value and a maximum value.

- DsDeSymbolicRule -- This is an abstract base class that provides the interface for creating concrete symbolic rules.
- DsDeSymbolicRuleFactory -- This class is used to create symbolic rules of the correct type.

Templates

- DsDeRealExpressionRule -- This template is used to specify an expression for validation.
- DsDeRealMatchRule -- This template is used to specify a set of valid values for an attribute.
- DsDeRealRangeRule -- This template is used to specify a valid range which includes a minimum value and a maximum value.

library DsDe2Sh (shared)

/ecs/formal/DSS/sdsrv/src/de2

This library contains the method to do validation, installing ESDT, sending metadata to IOS,DMS and SBSRV and generate MCF to ingest.

Classes

- DsDeAttNode -- Holds a single metadata attribute and provides services with respect to that attribute.
- DsDeConvert -- Used to do various types of conversions that are needed by the descriptor.
- DsDeDDAttribute -- Holds the dictionary information for DID 311 attributes.
- DsDeDDContent -- Holds the dictionary information for container objects.
- DsDeDictionary -- This class is used to hold the DID311 dictionary information.
- DsDeESDTCreator -- This class is used to add a new ESDT to the system.
- DsDeGroupNode -- Provides services with respect to the set of attributes.
- DsDeMCFBuilder -- Writes the MCF to a stream and also is capable of filtering out sections of the MCF.
- DsDeMCFtoODL -- Generates the MCF file in ODL.
- DsDeNode -- Holds any kind of metadata attribute part and is used to make arbitrary nested metadata constructs.
- DsDePSGroup -- Holds product specific metadata attributes.
- DsDePSNode -- Provides services for a single product specific metadata attribute.
- DsDeRealRule -- Provides the interface to creating concrete rules.

library DsGe

/ecs/formal/DSS/sdsrv/src/ge

This library provides the public interface for access to all ESDT services and ESDT specifications. It determines ESDT configuration information. It also performs runtime loading of ESDT libraries for a data type.

Classes

- DsGeDynamicLibrary -- Class that implements the dynamic library loading.
- DsGeECSDDataProduct -- Inherit from the public ESDT class, which provides basic ESDT functionality.
- DsGeESDT -- Inherit from the public class - provides basic ESDT functionality.
- DsGeESDTConfiguration -- This class is used to determine ESDT configuration information.
- DsGeESDTConfiguredType -- This struct is used to hold the information for a ESDT type.
- DsGeESDTDynamicLibrary -- This class performs runtime loading of ESDT libraries for a data type.
- DsGeESDTServiceProvider -- This abstract base class provides all of the truly public methods for ESDTs and ESDT Wrappers.
- DsGeESDTWrapper -- This class encapsulates the ESDT functionality and provides the public interface for access to all ESDT services.
- DsGeMetadataUpdate -- This class models the update of metadata, specifically for update during a subsetting operation on a ESDT object.
- DsGeOID -- This class provides a mechanism for uniquely identifying a data granule within a DataServer.
- DsGeScienceData -- Inherit from the public ESDT class, which provides basic ESDT functionality.

library DsGeSh (shared)

/ecs/formal/DSS/sdsrv/src/ge

This library provides the public interface for access to all ESDT services and ESDT specifications. It determines ESDT configuration information. It also performs runtime loading of ESDT libraries for a data type.

Classes

- DsGeDynamicLibrary -- Class that implements the dynamic library loading.
- DsGeECSDDataProduct -- Inherit from the public ESDT class, which provides basic ESDT functionality.
- DsGeESDT -- Inherit from the public class - provides basic ESDT functionality.
- DsGeESDTConfiguration -- This class is used to determine ESDT configuration information.
- DsGeESDTConfiguredType -- This struct is used to hold the information for a ESDT type.
- DsGeESDTDynamicLibrary -- This class performs runtime loading of ESDT libraries for a data type.
- DsGeESDTServiceProvider -- This abstract base class provides all of the truly public methods for ESDTs and ESDT Wrappers.
- DsGeESDTWrapper -- This class encapsulates the ESDT functionality and provides the public interface for access to all ESDT services.

- DsGeMetadataUpdate -- This class models the update of metadata, specifically for update during a subsetting operation on a ESDT object.
- DsGeOID -- This class provides a mechanism for uniquely identifying a data granule within a DataServer.
- DsGeScienceData -- Inherit from the public ESDT class, which provides basic ESDT functionality.

library DsHcSh (shared)

/ecs/formal/DSS/sdsrv/src/hc

This library provides a DCE-independent client interfaces for both Conformant and NonConformant subsetting. It also provide a client class derived from that can use Process Frame Work.

Classes

- DsHcClient -- This is the definition of HDF client that can use Process Frame Work.
- DsHcConformant -- Provides a DCE independent interface for Conformant Subsetting.
- DsHcNonConformant -- Provides a DCE independent interface for the NonConformant Subsetting.

library DsHrSh (shared)

/ecs/formal/DSS/sdsrv/src/hr

This library encapsulates all DCE exposure for the distributed objects and functions for both Conformant and NonConformant subsetting on the server side for HdfEos server. It also provides a client Proxy interface to the server's distributed objects. It contains the mechanisms for asynchronization and rebinding.

Classes

- DsHrAsynchRequest -- Contains the mechanisms of asynchronous requests to accomodate long requests such as WriteFile().
- DsHrConformantIDL_1_0
- DsHrConformantIDL_1_0_ABS
- DsHrConformantIDL_1_0_Mgr
- DsHrConformantProxy -- Provides a client interface to the server's distributed object..
- DsHrDCEConformantConcrete -- Encapsulates all DCE exposure for the connection objects on the server side.
- DsHrDCENonConfConcrete -- Encapsulates all DCE exposure for the connection objects on the server side.
- DsHrEnv -- This class provides the ability to check and get environment variables that are required buy HDF operations.
- DsHrEvent -- Represents an event on which the restart manager will work off.

- DsHrFhExecutor -- This is Fault handler class for DsHrConformantProxy and DsHrNonConfProxy.
- DsHrHdfEosListener -- This class is to compare the current memory size with the previous and according to some established policy, restart the HdfEos server or shut down the HdfEos server.
- DsHrListener -- The class is an abstract class which is instantiated by its derived class - DsHrHdfEosListener.
- DsHrManagedServer -- This class is derived from EcPfManagedServer and invoked by HDFEOS server.
- DsHrNonConfIDL_1_0
- DsHrNonConfIDL_1_0_ABS
- DsHrNonConfIDL_1_0_Mgr
- DsHrNonConfProxy -- Provides a client interface to the server's distributed object..
- DsHrQuit -- Server-side class that implements code to shutdown the Hdf Eos Server.
- DsHrQuitIDL_1_0
- DsHrQuitIDL_1_0_ABS
- DsHrQuitIDL_1_0_Mgr
- DsHrRestartManager -- Manages server restart upon receiving events and signals.
- DsHrTools -- Provides common functions needed by both Proxy and Concrete.

library DsLgSh (shared)

/ecs/formal/DSS/sdsrv/src/lg

This library provides a utility to hide the details of the Logging system.

library DsMd1Sh (shared)

/ecs/formal/DSS/sdsrv/src/md

This library defines metadata classes used to access a database.

Classes

- DsMdAttributeList
- DsMdCatalogBase -- This class is used to write entities to the Sybase database.
- DsMdCollDictionary -- This singleton class is used to cache the collection level metadata for ESDTs stored within the SDSRV.
- DsMdID -- This class serves as a wrapper class around the database ID.
- DsMdJoinTable -- This singleton class is used to store all the possible joins between the various tables in the database.
- DsMdPSTableAliasList -- This class is used to store the FROM clause elements associated with search for Product Specific Attributes within the database.

library DsMd2Sh (shared)

/ecs/formal/DSS/sdsrv/src/md

This library defines metadata classes used to access a database.

Classes

- DsMdCatalog
- DsMdCollDictionary -- This singleton class is used to cache the collection level metadata for ESDTs stored within the SDSRV.
- DsMdESDTConstraint -- This class contains the definitions for the DsMdESDTConstraint class.
- DsMdESDTConstraintArray
- DsMdGRingPoint -- This class supports GPolygon metadata insert by providing a class whose objects can be sorted by the "SequenceNumber" of the point.
- DsMdMetadata -- This class references to external ODL files.
- DsMdODL -- This class holds an ODL structure from an external file in memory and provides interfaces to manipulate this structure.

library DsMd3Sh (shared)

/ecs/formal/DSS/sdsrv/src/md

This library defines metadata classes used to access a database.

Classes

- DsMdAttr -- This class is not currently used by the system but is intended to help with porting from Sybase to other database systems.
- DsMdBaseDep -- Contains the list of tables to be processed as part of a metadata DeletePhysical.
- DsMdCommand -- Supports DeletePhysical by translating logical action names and table names to the corresponding Stored Procedure to execute.
- DsMdConvertToFromSQL -- To Take standard format input and convert to Database SQL commands.
- DsMdInsertSeq -- Contains the list of metadata lists to be processed as part of a metadata Insert.
- DsMdLookuptableBase -- A base class and derived singleton classes for maintaining lookup tables initially loaded from the database.
- DsMdOperator -- Translates from a logical operator name to the actual physical string required by the DBMS.
- DsMdUnloadList -- Contains the list of metadata lists to be processed as part of a metadata Insert.
- DsMdUpdateXrefList -- Contains the list of metadata lists to be processed as part of a metadata Update.

library DsSh

/ecs/formal/DSS/sdsrv/src/sh

This library provides underlying DCE(ODCE) mechanism and fundamental implementation for both client and server side of SDSRV. It also supplies utilities which is used by both client and server side code of SDSRV, such as cl, sr, ge, md, de libraries and etc.

Classes

- DsShCompression -- Defines the type of compression.
- DsShConnectionFhExecutor -- This is Fault handler class used for Submit() method in DsShConnectionProxy class.
- DsShConnectionIDL_4_0
- DsShConnectionIDL_4_0_ABS
- DsShConnectionIDL_4_0_Mgr
- DsShConnectionProxy -- This class provides a client interface to the server's connection distributed object.
- DsShConnectionReal -- This class provides a server interface to the server's connection distributed object. It inherits from the DCE-generated server connection class, and adds functions to provide stronger type-checking.
- DsShConVar -- Provides a condition variable class that encapsulates (and provides roughly the same functionality as) the underlying (currently DCE/threads) condition variable implementation.
- DsShDCEConnectionConcrete -- This class encapsulates all DCE exposure for the connection objects on the server-side.
- DsShDCEDescriptorConcrete -- This class encapsulates all of the DCE behavior and signatures for Descriptors and Descriptor collectors.
- DsShDCEId -- This class encapsulates the DCE object identifier, uuid_t.
- DsShDCEInterface -- This class encapsulates DCE implementations of DO interface functions, and adds other functions needed by other DCE implementation objects.
- DsShDCERef -- This class encapsulates the DCE object reference type DCEObjRefT.
- DsShDCESRequestConcrete -- This server-side class encapsulates all of the DCE behavior and signatures for srequests, and communicates with an associated DsShSRequestReal in a "DCE-clean" manner.
- DsShDCEUpdaterConcrete -- This class encapsulates all DCE exposure for the Updater objects on the server-side.
- DsShDescriptorFhExecutor -- This is Fault handler class used for Submit() method in DsShConnectionProxy class.
- DsShDescriptorIDL_4_0
- DsShDescriptorIDL_4_0_ABS
- DsShDescriptorIDL_4_0_Mgr

- DsShDescriptorProxy -- This class provides a DCE-independent interface to the Descriptor object and to the set of descriptor objects.
- DsShDescriptorReal -- This class provides a server interface to the server's request distributed object. It inherits from the DCE-generated server request class, and adds functions to provide stronger type-checking.
- DsShDOId -- Declaration of a distributed object (DO) id, independent of the underlying technology.
- DsShDOInterface -- This class provides a basic interface that all distributed object (DO) technologies must supply for the dataserver.
- DsShDORef -- This class encapsulates references to distributed objects in a manner that is independent of the underlying distributed object (DO) technology.
- DsShError -- This class is used for exceptions and error reporting in dataserver functions.
- DsShErrorDetails -- DsShErrorDetails is an accessor class for obtaining detailed information about an error code received from downstream.
- DsShESDTProvider -- This class is a shell provider for the ESDT UR.
- DsShESDTUR -- A UR is a persistent, transportable identifier that refers to C++ objects.
- DsShGenCommandImp -- This class is designed to be transferred from the client process space to the server process space, and is thus derived from RWCollectable, so that it can be "flattened" and "restored".
- DsShGenRequestImp -- Implementation class for requests.
- DsShGenRequestMediator -- A RequestMediator object will mediate interactions between DsShDCEConnectionConcrete (CC), DsShDCESRequestConcrete (RC), and DsShSRequestReal (RR) objects.
- DsShGenSessionManager -- The management of Sessions in the Sdsrv server should go into this class, including The creation and destruction of Sessions and accounting information.
- DsShLogging -- This class logs all DSS messages to a log, the definition of the severity level of messages logged, and a definition of the event types logged.
- DsShManagedClient -- DSS client which uses Process Frame Work .
- DsShMutex -- This class provides a mutex class that encapsulates the underlying (currently DCE) mutex implementation.
- DsShMutexLock -- This class provides a safe method of locking and unlocking DsShMutex objects, especially in the presence of exceptions.
- DsShNoPFTTestPro -- This is a the declaration of DsShNoPFTTestPro Class, which is solely for Test programs which do not have a PF initiated.
- DsShODLMutex -- This mutex will protect access to single thread ODL lib.
- DsShQuitIDL_4_0
- DsShQuitIDL_4_0_ABS
- DsShQuitIDL_4_0_Mgr
- DsShQuitProxy -- Client interface to the server shutdown class.

- DsShQuitReal -- This is a server-side class that implements code to shutdown the dataserver.
- DsShSciCommandImp -- This class specializes DsShGenCommandImp for the science data server to provide command categories.
- DsShSciRequestImp -- This class specializes DsShGenRequestImp for the science server, adding priorities and domains.
- DsShSciServerUR -- Universal Reference for binding information to Science Data Server.
- DsShServerUR -- Generic Server UR.
- DsShServerURProvider -- Provides URs to clients.
- DsShSRequestIDL_4_0
- DsShSRequestIDL_4_0_ABS
- DsShSRequestIDL_4_0_Mgr
- DsShSRequestProxy -- This class provides a client interface to the server's request distributed object, and translates the data types into the required DCE formats before calling the underlying DCE client-side request object..
- DsShSRequestReal -- This class provides a server interface to the server's request distributed object. It inherits from the DCE-generated server request class, and adds functions to provide stronger type-checking..
- DsShThread -- This class provides a thread class that encapsulates (and provides roughly the same functionality as) the underlying (currently DCE/threads) thread implementation.
- DsShTSSStorage -- This class provides a thread specific storage service.
- DsShTypeID -- This class represents a unique ESDT.
- DsShUpdaterIDL_4_0
- DsShUpdaterIDL_4_0_ABS
- DsShUpdaterIDL_4_0_Mgr
- DsShUpdaterProxy -- This class provides a client interface to the server's Updater distributed object.
- DsShUpdaterReal -- This class provides a server interface to the server's updater distributed object. It is primarily used in data server rundown condition while the client is making a asynchronous request with call back..

library DsShSh (shared)

/ecs/formal/DSS/sdsrv/src/sh

This internal library provides underlying DCE(ODCE) mechanism and fundamental implementation for both client and server side of SDSRV. It also supplies utilities which is used by both client and server side code of SDSRV, such as cl, sr, ge, md, de libraries and etc. The library has both static and shared versions. (unfortunately, due to historical reasons. some DsSh facilities are also used by other subsystems)

Classes

- DsShCompression -- Defines the type of compression.

- DsShConnectionFhExecutor -- This is Fault handler class used for Submit() method in DsShConnectionProxy class.
- DsShConnectionIDL_4_0
- DsShConnectionIDL_4_0_ABS
- DsShConnectionIDL_4_0_Mgr
- DsShConnectionProxy -- This class provides a client interface to the server's connection distributed object.
- DsShConnectionReal -- This class provides a server interface to the server's connection distributed object. It inherits from the DCE-generated server connection class, and adds functions to provide stronger type-checking.
- DsShConVar -- Provides a condition variable class that encapsulates (and provides roughly the same functionality as) the underlying (currently DCE/pthreads) condition variable implementation.
- DsShDCEConnectionConcrete -- This class encapsulates all DCE exposure for the connection objects on the server-side.
- DsShDCEDescriptorConcrete -- This class encapsulates all of the DCE behavior and signatures for Descriptors and Descriptor collectors.
- DsShDCEId -- This class encapsulates the DCE object identifier, uuid_t.
- DsShDCEInterface -- This class encapsulates DCE implementations of DO interface functions, and adds other functions needed by other DCE implementation objects.
- DsShDCERef -- This class encapsulates the DCE object reference type DCEObjRefT.
- DsShDCESRequestConcrete -- This server-side class encapsulates all of the DCE behavior and signatures for srequests, and communicates with an associated DsShSRequestReal in a "DCE-clean" manner.
- DsShDCEUpdaterConcrete -- This class encapsulates all DCE exposure for the Updater objects on the server-side.
- DsShDescriptorFhExecutor -- This is Fault handler class used for Submit() method in DsShConnectionProxy class.
- DsShDescriptorIDL_4_0
- DsShDescriptorIDL_4_0_ABS
- DsShDescriptorIDL_4_0_Mgr
- DsShDescriptorProxy -- This class provides a DCE-independent interface to the Descriptor object and to the set of descriptor objects.
- DsShDescriptorReal -- This class provides a server interface to the server's request distributed object. It inherits from the DCE-generated server request class, and adds functions to provide stronger type-checking.
- DsShDOId -- Declaration of a distributed object (DO) id, independent of the underlying technology.
- DsShDOInterface -- This class provides a basic interface that all distributed object (DO) technologies must supply for the dataserver.

- DsShDORef -- This class encapsulates references to distributed objects in a manner that is independent of the underlying distributed object (DO) technology.
- DsShError -- This class is used for exceptions and error reporting in dataserver functions.
- DsShErrorDetails -- DsShErrorDetails is an accessor class for obtaining detailed information about an error code received from downstream.
- DsShESDTProvider -- This class is a shell provider for the ESDT UR.
- DsShESDTUR -- A UR is a persistent, transportable identifier that refers to C++ objects.
- DsShGenCommandImp -- This class is designed to be transferred from the client process space to the server process space, and is thus derived from RWCollectable, so that it can be "flattened" and "restored".
- DsShGenRequestImp -- Implementation class for requests.
- DsShGenRequestMediator -- A RequestMediator object will mediate interactions between DsShDCEConnectionConcrete (CC), DsShDCESRequestConcrete (RC), and DsShSRequestReal (RR) objects.
- DsShGenSessionManager -- The management of Sessions in the Sdsrv server should go into this class, including The creation and destruction of Sessions and accounting information.
- DsShLogging -- This class logs all DSS messages to a log, the definition of the severity level of messages logged, and a definition of the event types logged.
- DsShManagedClient -- DSS client which uses Process Frame Work .
- DsShMutex -- This class provides a mutex class that encapsulates the underlying (currently DCE) mutex implementation.
- DsShMutexLock -- This class provides a safe method of locking and unlocking DsShMutex objects, especially in the presence of exceptions.
- DsShNoPFTestPro -- This is a the declaration of DsShNoPFTestPro Class, which is solely for Test programs which do not have a PF initiated.
- DsShODLMutex -- This mutex will protect access to single thread ODL lib.
- DsShQuitIDL_4_0
- DsShQuitIDL_4_0_ABS
- DsShQuitIDL_4_0_Mgr
- DsShQuitProxy -- Client interface to the server shutdown class.
- DsShQuitReal -- This is a server-side class that implements code to shutdown the dataserver.
- DsShSciCommandImp -- This class specializes DsShGenCommandImp for the science data server to provide command categories.
- DsShSciRequestImp -- This class specializes DsShGenRequestImp for the science server, adding priorities and domains.
- DsShSciServerUR -- Universal Reference for binding information to Science Data Server.
- DsShServerUR -- Generic Server UR.
- DsShServerURProvider -- Provides URs to clients.

- DsShSRequestIDL_4_0
- DsShSRequestIDL_4_0_ABS
- DsShSRequestIDL_4_0_Mgr
- DsShSRequestProxy -- This class provides a client interface to the server's request distributed object, and translates the data types into the required DCE formats before calling the underlying DCE client-side request object..
- DsShSRequestReal -- This class provides a server interface to the server's request distributed object. It inherits from the DCE-generated server request class, and adds functions to provide stronger type-checking..
- DsShThread -- This class provides a thread class that encapsulates (and provides roughly the same functionality as) the underlying (currently DCE/pthreads) thread implementation.
- DsShTSSStorage -- This class provides a thread specific storage service.
- DsShTypeID -- This class represents a unique ESDT.
- DsShUpdaterIDL_4_0
- DsShUpdaterIDL_4_0_ABS
- DsShUpdaterIDL_4_0_Mgr
- DsShUpdaterProxy -- This class provides a client interface to the server's Updater distributed object.
- DsShUpdaterReal -- This class provides a server interface to the server's updater distributed object. It is primarily used in data server rundown condition while the client is making a asynchronous request with call back..

library DsSiSh (shared)

/ecs/formal/DSS/sdsrv/src/si

The DsSi library provides database support for the persistent system information including system requests and service commands.

Classes

- DsSiMemento
- DsSiMemMonitor -- This class is used to reserve and monitor free memory conditions within the process.
- DsSiSysDataCatalog

library DsSrSh (shared)

/ecs/formal/DSS/sdsrv/src/sr

This is an internal library for Science Data Server. It provides some functions which are used by other directories's programs. These functions include reading configuration file, get access to metadata database and get access to HDFEOS server and so on.

Classes

- DsSrConnectionMaker -- Provides a specialized NewConnection function for DsShDCEConnectionConcrete to create DsSrSession objects.
- DsSrDescriptorMaker -- Provides specialized services for creation of server side descriptor objects.
- DsSrEnv -- This class provides the ability to check and get environment variables that are required by DSS operations.
- DsSrGenCatalogPool -- Used by Science Data Server objects to obtain access to a catalog.
- DsSrGenCatalogWrapper -- Used by data server objects to obtain access to a catalog.
- DsSrGenConnection -- Distributed base class for all connections/sessions that exist in Science Data Server.
- DsSrGenQueuedConnection -- Provides the ability to queue requests as they arrive.
- DsSrGenRequestInt -- Server side object that represents a request to the data server.
- DsSrHDFServerPool -- Used by the Science Data Server to obtain access to the HDF EOS Servers.
- DsSrHDFServerToken -- Used by the Science Data Server to obtain access to the HDF EOS Servers.
- DsSrManagedServer -- Hides the interface between MSS and the application by instantiating the EcAgManager object.
- DsSrRequest -- Server side object that represents a request to the Science Data Server.
- DsSrSession -- A specialization of the DsSrGenQueuedConnection class that adds DsSrWorkingCollection-oriented behavior to the connection.
- DsSrSysCatalogPool -- Used by Science Data Server objects to obtain access to a System Catalog.
- DsSrSysCatalogWrapper -- Used by Science Data Server objects to obtain access to a System Catalog.
- DsSrWarmStartManager -- Processes the unfinished asynchronous acquire requests that were left over the last time the Science Data Server was operating.
- DsSrWorkingCollection -- Executes commands by category.

library DsSt8MMServerSh (shared)

/ecs/formal/DSS/stmgt/src/media/server

This directory contains the source code for all media servers, including the D3 server, the 8MM server, the DLT server, and the CDROM server. The source code creates both libraries and executables for the Sun platform.

Classes

- DsSt8MMReal
- DsSt8MMServiceThread

library DsStArClientAsyncSh (shared)

/ecs/formal/DSS/stmgt/src/archive

This library libDsStArClientAsync is used by asynchronous clients (ie other STMGT Servers) of the Archive Server to perform archive retrieves and archive stores.

Classes

- DsStArchiveAsyncIF -- onstructor.
- DsStArchiveRequest -- CREATE DATE: 2/16/00.
- DsStArchiveRestoreParams
- DsStArchiveResult -- CREATE DATE: 2/16/00.
- DsStFileParameters -- Provides basic CSDT functionality.

library DsStArClientSync

/ecs/formal/DSS/stmgt/src/archive

This library libDsStArClientSync is used by synchronous clients of the Archive Server to perform archive retrieves and archive stores.

Classes

- DsStArchive
- DsStArchiveRequest -- CREATE DATE: 2/16/00.
- DsStArchiveRestoreParams
- DsStArchiveResult -- CREATE DATE: 2/16/00.
- DsStFileParameters -- Provides basic CSDT functionality.

library DsStArClientSyncSh (shared)

/ecs/formal/DSS/stmgt/src/archive

This library libDsStArClientSync is used by synchronous clients of the Archive Server to perform archive retrieves and archive stores.

Classes

- DsStArchive
- DsStArchiveRequest -- CREATE DATE: 2/16/00.
- DsStArchiveRestoreParams
- DsStArchiveResult -- CREATE DATE: 2/16/00.
- DsStFileParameters -- Provides basic CSDT functionality.

library DsStArServerSh (shared)

/ecs/formal/DSS/stmgt/src/archive

This library libDsStArServer is used by the ArchiveServer to perform its core functions.

Classes

- DsStArchiveDBIF -- This class interfaces with DsStCommonDBIF to perform DB queries.
- DsStArchiveReadReal
- DsStArchiveReadThread
- DsStArchiveReal -- This class provides a server interface to the server's request distributed object. It inherits from the DCE-generated server request class, and adds functions to provide stronger type-checking.
- DsStArchiveRPCInfo
- DsStArchiveServiceThread
- DsStArchiveWriteReal
- DsStArchiveWriteThread
- DsStFileParameters -- Provides basic CSDT functionality.

library DsStCDROMDeviceServiceSh (shared)

/ecs/formal/DSS/stmgt/src/media/service/device

This directory contains source code for service classes which provide access to devices, both for standalone and stacker-based devices. The source code produces libraries for the Sun platform.

Classes

- DsStCDROMDeviceService

library DsStCDROMServerSh (shared)

/ecs/formal/DSS/stmgt/src/media/server

This directory contains the source code for all media servers, including the D3 server, the 8MM server, the DLT server, and the CDROM server. The source code creates both libraries and executables for the Sun platform.

Classes

- DsStCDROMReal
- DsStCDROMServiceThread -- , DsStMediaStagingServiceThread.

library DsStCDROMServiceSh (shared)

/ecs/formal/DSS/stmgt/src/media/service/media

This directory contains source code for service classes which provide access to media, both for media distribution and media ingest purposes. The source code produces libraries for the Sun platform.

Classes

- DsStCDROMService
- DsStKparCDIF -- Callback function needed for monitor progress.
- DsStMediaService -- Class DsStMediaService works as a root class to enable that all overriden member functions in the subclasses can be polymorphically referenced in DsStMediaserviceThred level.

library DsStCmClientSh (shared)

/ecs/formal/DSS/stmgt/src/cache

The library libDsStCmClient is used by CacheManager clients (EcDsStArchiveServer & EcDsStFtpServer) to access the functionality of the CacheManager Server.

Classes

- DsStCacheManagerAsyncIF -- 2/8/00.
- DsStCacheManagerRequest -- CREATE DATE: 2/8/00.
- DsStCacheManagerResult -- CREATE DATE: 2/8/00.

library DsStCmnDb

/ecs/formal/DSS/stmgt/src/cmn/db

The library libDsStCmnDb is used by STMGT Servers to access core database functionality including database connection, stored procedure execution and database transaction processing.

Classes

- DsStCommonDBIF -- If a global instance is needed which is outside of a class declare the variable static.
- DsStDbConnection
- DsStDbConnectionManager
- DsStDbInterface -- Database (Sybase) interface class to encapsulate database related services such as: connect, execute, fetch result.
- DsStDictionary -- Data dictionary that maps a string name to a string value.
- DsStParameter -- Represents a value of a parameter.
- DsStParameterList
- DsStStoredProcedures
- DsStStProcTable -- This is a singleton used to create a dictionary that associates a stored procedure name with a list of parameters in the order that they appear in the stored procedure declaration.

library DsStCmnDbSh (shared)

/ecs/formal/DSS/stmgt/src/cmn/db

The library libDsStCmnDb is used by STMGT Servers to access core database functionality including database connection, stored procedure execution and database transaction processing.

Classes

- DsStCommonDBIF -- If a global instance is needed which is outside of a class declare the variable static.
- DsStDbConnection
- DsStDbConnectionManager
- DsStDbInterface -- Database (Sybase) interface class to encapsulate database related services such as: connect, execute, fetch result.
- DsStDictionary -- Data dictionary that maps a string name to a string value.
- DsStParameter -- Represents a value of a parameter.
- DsStParameterList
- DsStStoredProcedures
- DsStStProcTable -- This is a singleton used to create a dictionary that associates a stored procedure name with a list of parameters in the order that they appear in the stored procedure declaration.

library DsStCmnDCESh (shared)

/ecs/formal/DSS/stmgt/src/cmn/dce

The library libDsStCmnDCE is used by STMGT Servers to access core DCE functionality.

Classes

- DsStBaseCFhExecutor -- All custom stmgt executors will inherit from this class.
- DsStDCEServer -- Security.
- DsStServerConcrete -- Declarations for DsStServerConcrete class which is a pure virtual base class used to invoke the Startup() method.

library DsStCmnFileIO

/ecs/formal/DSS/stmgt/src/cmn/fileio

The library libDsStCmnFileIO is used by STMGT Servers to access core io functionality including checksumming, copying and compression.

Classes

- DsStCheckSumService -- Class declaration to calculate check sum.
- DsStCompressCreator -- This class is to return the object of the type of compression requested.

- DsStCopyService -- Class declaration to copy files.
- DsStFileCompress -- This class is to compress files.
- DsStUnixCompress -- This class is used to do Unix compression.
- DsStUtility

library DsStCmnFileIOSh (shared)

/ecs/formal/DSS/stmgt/src/cmn/fileio

The library libDsStCmnFileIO is used by STMGT Servers to access core io functionality including checksumming, copying and compression.

Classes

- DsStCheckSumService -- Class declaration to calculate check sum.
- DsStCompressCreator -- This class is to return the object of the type of compression requested.
- DsStCopyService -- Class declaration to copy files.
- DsStFileCompress -- This class is to compress files.
- DsStUnixCompress -- This class is used to do Unix compression.
- DsStUtility

library DsStCmnLogging

/ecs/formal/DSS/stmgt/src/cmn/logging

The library libDsStCmnLogging is used by STMGT Servers to access error code and logging functionality.

Classes

- DsStErrorDetails -- DsStErrorDetails is an accessor class for obtaining detailed information about an error code received from downstream.
- DsStErrorMsg -- "..
- DsStLogging

library DsStCmnLoggingSh (shared)

/ecs/formal/DSS/stmgt/src/cmn/logging

The library libDsStCmnLogging is used by STMGT Servers to access error code and logging functionality.

Classes

- DsStErrorDetails -- DsStErrorDetails is an accessor class for obtaining detailed information about an error code received from downstream.
- DsStErrorMsg -- "..

- DsStLogging

library DsStCmnServerSh (shared)

/ecs/formal/DSS/stmgt/src/cmn/server

The library libDsStCmnServer is used by STMGT Servers to access core server functionality including database access parameters and thread configuration.

Classes

- DsStProcess
- DsStTestDriver

library DsStCmServerSh (shared)

/ecs/formal/DSS/stmgt/src/cache

The library libDsStCMServer is used by the CacheManager Server to access core functionality.

Classes

- DsStCacheManagerReal
- DsStCacheManagerRPCInfo
- DsStCacheManagerServiceThread -- 2/8/00.

library DsStD3ServerSh (shared)

/ecs/formal/DSS/stmgt/src/media/server

This directory contains the source code for all media servers, including the D3 server, the 8MM server, the DLT server, and the CDROM server. The source code creates both libraries and executables for the Sun platform.

Classes

- DsStD3Real
- DsStD3ServiceThread

library DsStDeviceServiceSh (shared)

/ecs/formal/DSS/stmgt/src/media/service/device

This directory contains source code for service classes which provide access to devices, both for standalone and stacker-based devices. The source code produces libraries for the Sun platform.

Classes

- DsStDeviceService
- DsStStackerDBIF
- DsStStackerLib
- DsStStackerService

library DsStDLTServerSh (shared)

/ecs/formal/DSS/stmgt/src/media/server

This directory contains the source code for all media servers, including the D3 server, the 8MM server, the DLT server, and the CDROM server. The source code creates both libraries and executables for the Sun platform.

Classes

- DsStDLTReal
- DsStDLTServiceThread

library DsStFtp

/ecs/formal/DSS/stmgt/src/ftp/service

The library libDsStFtp is used by STMGT Servers to perform file transfers using ftp.

Classes

- DsStFtpService -- Library for ftpd clients.(libftp).

library DsStFtpClientSync

/ecs/formal/DSS/stmgt/src/ftp/client

This directory contains the source code for the client interface to the FTP Server. Note that one piece of the client interface is shared with the media servers (). That piece may be found in the media/client directory. This source code creates libraries for both Sun and SGI platforms.

Classes

- DsStFtpClient
- DsStFtpRequest
- DsStFtpRestoreParams
- DsStFtpResult

library DsStFtpClientSyncSh (shared)

/ecs/formal/DSS/stmgt/src/ftp/client

This directory contains the source code for the client interface to the FTP Server. Note that one piece of the client interface is shared with the media servers (). That piece may be found in the media/client directory. This source code creates libraries for both Sun and SGI platforms.

Classes

- DsStFtpClient
- DsStFtpRequest
- DsStFtpRestoreParams
- DsStFtpResult

library DsStFtpServerSh (shared)

/ecs/formal/DSS/stmgt/src/ftp/server

The code in the disftp directory provides the server side code for handling distribution ftp requests. The server utilizes the CSS provided ftp interface to provide the ftp push services. The ftp pull services also utilize the CSS provided interface to ftp in order to push the files to the pull cache area where they are monitored by the Pull Monitor server (code found in the pull subdirectory under src). This code will run on both Sun and SGI platforms.

Classes

- DsStFtpReal
- DsStFtpRPCInfo -- This class defines the attributes associated with FTP requests.
- DsStFtpServiceThread

library DsStFtpSh (shared)

/ecs/formal/DSS/stmgt/src/ftp/service

The library libDsStFtp is used by STMGT Servers to perform file transfers using ftp.

Classes

- DsStFtpService -- Library for ftpd clients.(libftp).

library DsStMediaClientAsyncSh (shared)

/ecs/formal/DSS/stmgt/src/media/client

Classes

- DsStMediaAsyncIF
- DsStMediaRequest
- DsStMediaRestoreParams
- DsStMediaResult -- This class serves to get the results of a request from the database.
- DsStStackerRequest
- DsStStackerResult -- This class serves to get the results of a request from the database.

library DsStMediaClientSync

/ecs/formal/DSS/stmgt/src/media/client

Classes

- DsStMediaClient
- DsStMediaRequest
- DsStMediaRestoreParams
- DsStMediaResult -- This class serves to get the results of a request from the database.

- DsStResourceProvider
- DsStStackerRequest
- DsStStackerRestoreParams
- DsStStackerResult -- This class serves to get the results of a request from the database.

library DsStMediaClientSyncSh (shared)

/ecs/formal/DSS/stmgt/src/media/client

Classes

- DsStMediaClient
- DsStMediaRequest
- DsStMediaRestoreParams
- DsStMediaResult -- This class serves to get the results of a request from the database.
- DsStResourceProvider
- DsStStackerRequest
- DsStStackerRestoreParams
- DsStStackerResult -- This class serves to get the results of a request from the database.

library DsStMediaServersSh (shared)

/ecs/formal/DSS/stmgt/src/media/server

This directory contains the source code for all media servers, including the D3 server, the 8MM server, the DLT server, and the CDROM server. The source code creates both libraries and executables for the Sun platform.

Classes

- DsStMediaRPCInfo -- This class defines the attributes associated with Media requests.
- DsStMediaServiceThread -- "..

library DsStMediaServiceSh (shared)

/ecs/formal/DSS/stmgt/src/media/service/media

This directory contains source code for service classes which provide access to media, both for media distribution and media ingest purposes. The source code produces libraries for the Sun platform.

Classes

- DsStMediaService -- Class DsStMediaService works as a root class to enable that all overriden member functions in the subclasses can be polymopohically referenced in DsStMediaserviceThred level.
- DsStStackerDBIF

- DsStTapeService
- DsStTarFormat -- "..

library DsStMediaStagingSh (shared)

/ecs/formal/DSS/stmgt/src/media/staging

This directory contains source code to support local staging of data for all media servers. The source code creates libraries for the Sun platform.

Classes

- DsStMediaStagingReal
- DsStMediaStagingServiceThread

library DsStRmClient

/ecs/formal/DSS/stmgt/src/reqmgr

The library libDsStRmClient is used by STMGT clients to access STMGT servers through the DCE Interface.

Classes

- DsStReqMgrIDL_4_0
- DsStRequestManager -- This is the base class for all client interfaces to STMGT servers.
- DsStRequestManagerProxy -- This is the base class for all client-side DCE interfaces to STMGT servers.

library DsStRmClientSh (shared)

/ecs/formal/DSS/stmgt/src/reqmgr

The library libDsStRmClient is used by STMGT clients to access STMGT servers through the DCE Interface.

Classes

- DsStReqMgrIDL_4_0
- DsStRequestManager -- This is the base class for all client interfaces to STMGT servers.
- DsStRequestManagerProxy -- This is the base class for all client-side DCE interfaces to STMGT servers.

library DsStRmServerSh (shared)

/ecs/formal/DSS/stmgt/src/reqmgr

The library lib_DsStRmServer is used by the Request Manager to access core request functionality. This includes thread management and request processing.

Classes

- DsStHousekeeper -- This class provides a thread to check for completed requests.
- DsStReqMgrIDL_4_0_ABS
- DsStReqMgrIDL_4_0_Mgr
- DsStRequest
- DsStRequestManagerConcrete
- DsStRequestManagerReal
- DsStRequestManagerServiceThread
- DsStRequestQueue

library DsStSCHServiceSh (shared)

/ecs/formal/DSS/stmgt/src/media/service/device

This directory contains source code for service classes which provide access to devices, both for standalone and stacker-based devices. The source code produces libraries for the Sun platform.

Classes

- DsStSCHStackerLib

library DsStSdClientAsyncSh (shared)

/ecs/formal/DSS/stmgt/src/sdisk

The library libDsStSdClientASync is used by the clients of the Staging Disk Server to allocate and attach to staging disks, copy and link files in staging disks and to create subdirectories in staging disks.

Classes

- DsStStagingDiskAsyncIF
- DsStStagingDiskFile
- DsStStagingDiskRequest -- CREATE DATE: 3/02/00.
- DsStStagingDiskRestoreParams -- CREATE DATE: 3/03/00.
- DsStStagingDiskResult -- CREATE DATE: 3/02/00.

library DsStSdClientSync

/ecs/formal/DSS/stmgt/src/sdisk

The library libDsStSdClientSync is used by the clients of the Staging Disk Server to allocate and attach to staging disks, copy and link files in staging disks and to create subdirectories in staging disks.

Classes

- DsStStagingDisk -- DsStStagingDisk class allows access to files and directories on the working storage disk space managed by the storage management software.
- DsStStagingDiskController -- This class allows another class to update the Staging Disk remaining space and have access to the path for the staging Disk Users of this class take explicit responsibility for the updates to the size of the staging disk.
- DsStStagingDiskFile
- DsStStagingDiskRequest -- CREATE DATE: 3/02/00.
- DsStStagingDiskRestoreParams -- CREATE DATE: 3/03/00.
- DsStStagingDiskResult -- CREATE DATE: 3/02/00.

library DsStSdClientSyncSh (shared)

/ecs/formal/DSS/stmgt/src/sdisk

The library libDsStSdClientSync is used by the clients of the Staging Disk Server to allocated and attach to staging disks, copy and link files in staging disks and to create subdirectories in staging disks.

Classes

- DsStStagingDisk -- DsStStagingDisk class allows access to files and directories on the working storage disk space managed by the storage management software.
- DsStStagingDiskController -- This class allows another class to update the Staging Disk remaining space and have access to the path for the staging Disk Users of this class take explicit responsibility for the updates to the size of the staging disk.
- DsStStagingDiskFile
- DsStStagingDiskRequest -- CREATE DATE: 3/02/00.
- DsStStagingDiskRestoreParams -- CREATE DATE: 3/03/00.
- DsStStagingDiskResult -- CREATE DATE: 3/02/00.

library DsStSdServerSh (shared)

/ecs/formal/DSS/stmgt/src/sdisk

This directory contains the source code for the Staging Disk Server. Both client interface code and the server itself are located in this directory. The source code creates both libraries and executables for the Sun and SGI platforms.

Classes

- DsStStagingDiskReal
- DsStStagingDiskRPCInfo
- DsStStagingDiskServiceThread -- 3/20/00.
- DsStStagingDiskTable -- The DsStStagingDiskTable class is the interface to the staging disk table and the inventory table in the database.

library DsStSRIServiceSh (shared)

/ecs/formal/DSS/stmgt/src/media/service/device

This directory contains source code for service classes which provide access to devices, both for standalone and stacker-based devices. The source code produces libraries for the Sun platform.

Classes

- DsStSRISStackerLib

library DsStStandaloneServiceSh (shared)

/ecs/formal/DSS/stmgt/src/media/service/device

This directory contains source code for service classes which provide access to devices, both for standalone and stacker-based devices. The source code produces libraries for the Sun platform.

Classes

- DsStStandaloneService

library DsStTmClientSh (shared)

/ecs/formal/DSS/stmgt/src/threadmgr

The library libDsStTmClientSh is used by STMGT Servers that are asynchronous clients of other STMGT Servers.

Classes

- DsStBaseAsyncIF
- DsStPatron -- DsStPatron's sole reason for existence is to provide a static method, `AwakenRemoteServer`, which attempts to make a connection to a server. The connection awakens a process begun by the `DsStReceptionist` constructor..

library DsStTmServerSh (shared)

/ecs/formal/DSS/stmgt/src/threadmgr

The library libDsStTmServerSh is used by STMGT Servers to access core functionality for managing service threads, manager threads and request information.

Classes

- DsStBaseReal -- This is the base class for all STMGT servers.
- DsStReceptionist -- This class provides the common mechanism for one server to awaken another, where both servers are based on the thread manager architecture.
- DsStRPCInfo -- This class defines the attributes associated with any request in Storage Management.
- DsStServiceThread

library DsStTmShared

/ecs/formal/DSS/stmgt/src/threadmgr

The library libDsStTmSharedSh is used by STMGT Servers to access functionality necessary to process remote and local transactions.

Classes

- DsStDBIFTransaction
- DsStRemoteSP
- DsStRemoteTransaction
- DsStResultSet
- DsStResultValue

library DsStTmSharedSh (shared)

/ecs/formal/DSS/stmgt/src/threadmgr

This directory contains the source code for the Thread Manager architecture which serves as the underlying design pattern for all STMGT servers. The source code creates several libraries and test executables for both the Sun and SGI platforms.

Classes

- DsStDBIFTransaction
- DsStRemoteSP
- DsStRemoteTransaction
- DsStResultSet
- DsStResultValue

library DsTlSh (shared)

/ecs/formal/DSS/sdsrv/src/tl

This library provides typed safe (smart) pointers / objects for RogueWave classes and a utility to make logging complex error messages easier.

Classes

- DsTlMsg

library DsUtSh (shared)

/ecs/formal/DSS/sdsrv/src/ut

This library provides utilities to support granule deletion functionality.

Classes

- DsUtCommandLine -- This class contains methods for class DsUtCommandLine.

- DsUtDeleteManager -- Contains methods for class DsUtDeleteManager.
- DsUtGranuleDelete -- This header contains all the methods for DsUtGranuleDelete.

library EcAgInstrm

/ecs/formal/MSS/MACI/src

This library contains the instrumentation code which is linked into all ECS managed servers.

The instrumentation code enables the managed servers to: - find the CDS path to the subagent and register with the subagent upon startup - send events to the subagent - register metrics with the subagent - shutdown gracefully

Classes

- EcAgConfigFile -- This class embodies the characteristics and operations of the management configuration file.
- EcAgConfigMetric -- This metric contains configuration data.
- EcAgEvent -- EcAgEvent is not an abstract class.
- EcAgException -- Binds error messages to DCE return codes.
- EcAgFaultMetric -- This metric contains fault data.
- EcAgFileMgr -- This class provides much the same functionality as EcAgRcvFile except that it doesn't inherit from rwFile. It allows for certain types of file recovery..
- EcAgHostInfo -- This class provides the additional information associated with an event, needed by the logger and the deputy.
- EcAgLockableFile -- This class encapsulates the stdio FILE* struct, encapsulating the UNIX file locking mechanism.
- EcAgManager -- The EcAgManager is an OODCE distributed object.
- EcAgManagerOD_1_0_ABS
- EcAgManagerOD_1_0_Mgr
- EcAgMetric -- This is an abstract class that defines the common operations such as (un-) Lock, save/restoreGuts.
- EcAgNamedList -- This class represents a RW singly linked list that has a text name.
- EcAgPerfMetric -- This metric contains performance data.
- EcAgRcvFile -- This is a specialized version of RWFile which contains recoverability capabilities.
- EcAgResettingPerfMetric -- This metric contains performance data.
- EcAgTuple -- This class represents a tuple which will be stored in an EcAgEvent.
- MsAgCfgFileInfo -- This class is used to put into a temporary vector so that when MsAgDiscoverer compares what's installed and what's in the subagent's table, it can do this easily..
- MsAgCPU -- This class retrieves the percentage of CPU usage of a process.
- MsAgEventHandler -- This class handles event for EcAgManager.

- MsAgEventManagerOD_1_0
- MsAgInstGen -- This class generates a unique instance ID.
- MsAgIntConfigMetric -- This class is a specialized class of EcAgConfigMetric.
- MsAgLockableObject -- The first class is used to lock objects such as the lockable queue.
- MsAgLockableQueue -- Lockable queue object inherits from lockable object and rogue wave collectables queue.
- MsAgLockableSList
- MsAgLockableSortedVector
- MsAgMetricChecker
- MsAgMetricHandler -- This class handles all the reading and creating of metrics for EcAgManager.
- MsAgMetVector -- This class is used to store EcAgMetrics in an RWOrdered array.
- MsAgPathFinder -- This function builds the path and filename of subagent related files and CDS names.
- MsAgProcess -- This file determines if a process is still alive.
- MsAgProcInfo -- This class provides a mechanism to retrieve information from the system.
- MsAgProcPerfMetric -- This class is a special performance metric that contains either cpu, memory, various RPC information, the number of threads, or the number of disk I/O.
- MsAgProcSShotInfo -- This class provides a mechanism to retrieve information from the same snap-shot for the classes that inherited from the performance metric class of the EcAgManager.
- MsAgRegistryOD_1_0
- MsAgUpdateConfigMetric -- This class is a specialized class of MsAgUpdateCfgMetric, which is tied directly with the update interval seconds for MsAgProcSShotInfo.

library EcCf

/ecs/formal/CSS/DOF/src/CF/cf

The Config library (CF) is a static library provides the necessary software for the PF_Client & the PF_Server applications define their own set of configuration related information in a single location and retrieve them at the application startup.

Classes

- EcPfConfigFile -- This class is used to read the Configuration file that holds the values of attributes needed for the process Framework.
- ListAndNameCollect -- This class is used to create collectable nodes whose fields are the attribute name and the linked list of the values for the respective attribute.

library EcCfSh (shared)

/ecs/formal/CSS/DOF/src/CF/cf

The Config library (CF) is a shared library provides the necessary software for the PF_Client & the PF_Server applications define their own set of configuration related information in a single location and retrieve them at the application startup.

Classes

- EcPfConfigFile -- This class is used to read the Configuration file that holds the values of attributes needed for the process Framework.
- ListAndNameCollect -- This class is used to create collectable nodes whose fields are the attribute name and the linked list of the values for the respective attribute.

library EcCIWbCgiVar

/ecs/formal/CLS/WKBCH/HtmlLib

This library contains utilities to read HTML templates and build HTML pages from the templates. It also provides classes that encapsulate the values that can be read into a CGI program.

Classes

- CIWbCgiElement
- CIWbCgiVars
- CIWbTplElement
- CIWbTplVars

library EcCsGateway

/ecs/formal/CSS/DOF/src/GATEWAY/src/common

This static library may be used to send/receive gateway messages securely at a user specified level. Currently, only the kerberos security is provided. It also stores and processes the messages that are passed to and from the gateway. There is also functionality included to aid in processing messages for the SPDF Gateway.

Classes

- CsGwGateway -- This class may be used to send/receive gateway messages securely at a user specified level.
- CsGwMessage -- This class may be used to store and process the messages passing to and from Rel A communication gateway.
- CsGwSdpfMsg -- This derived class adds more functionalities to the CsGwMessage class as needed in processing messages for SDPF gateway.

library EcCsMtMGateway

/ecs/formal/CSS/DOF/src/RELB_GATEWAY/MTMGW/src

Classes

- EcCsMtMAcctSrvMgr -- This class encapsulates the services of ECS Account Management Server.
- EcCsMtMAttributeDict
- EcCsMtMConn -- This class encapsulates a structure that is established for each connection event.
- EcCsMtMDaemonState -- EcCsMtMDaemonState holds information pertinent to the daemon as a whole.
- EcCsMtMDataServerMgr -- This class is an abstraction wrapping up all the operations Data Server provides.
- EcCsMtMECSOrderProxy -- This class works as a surrogate for MTMGW in handling order request The public interface ProcessRequest() overriding that in parent class takes care of order request.
- EcCsMtMECSProxy -- Define a virtual interface by which the various ECS service facilities are invoked.
- EcCsMtMECSSearchOrderProxy -- This class works as a surrogate for MTMGW in handling search and order request.
- EcCsMtMECSSearchProxy -- This class works as a surrogate for MTMGW in handling search request, The public interface ProcessRequest() overriding that in parent class takes care of search request.
- EcCsMtMGIToXMLConverter -- This class is a concrete class to EcCsMtMMsgConverter.
- EcCsMtMManagedSrv -- This is an application of the Process Framework(PF).
- EcCsMtMMessage -- This class is used as parsing class.
- EcCsMtMMessageCatalog -- EcCsMtMMessageCatalog is a singleton object which provides the look up table for the error code and its message used through out the MTM Gateway.
- EcCsMtMOption -- These are options set from the server command line or configuration file or whatever.
- EcCsMtMOrderImp -- This class handles the concrete order implementation that could be easily coupled or decoupled with its interface, here which is EcCsMtMECSOrderProxy.
- EcCsMtMProxy -- EcCsMtMProxy defines an interface by which the various ECS service facilities are invoked.
- EcCsMtMProxyCreator -- EcCsMtMProxyCreator defines an interface for creating an object of type EcCsMtMProxy, but let subclasses decide which class to instantiate.
- EcCsMtMSdsrvMgr -- This class is a derived class of EcCsMtMDataServerMgr.
- EcCsMtMSearchImp -- This class handles the concrete search implementation that could be easily coupled or decoupled with its interface, where with is EcCsMtMECSSearchProxy.

- EcCsMtMSearchOrderImp -- This class handles the concrete search and order request implementation that could be easily coupled or decoupled with its interface, here which is EcCsMtMECSearchOrderProxy.
- EcCsMtMSocketSrv -- This class does the basic socket listen operation.

library EcCsRegistry

/ecs/formal/CSS/DOF/src/REGISTRY/registry

This library retrieves configuration attribute-value pairs and distribution options from the Configuration Registry Database.

Classes

- EcRgDistOptions -- This class requests the distribution media options from the EcCsRegistryServer.
- EcRgRegistry_1_0
- EcRgRegistryServer_C

library EcCsRegistrySh (shared)

/ecs/formal/CSS/DOF/src/REGISTRY/registry

Classes

- EcRgDistOptions -- This class requests the distribution media options from the EcCsRegistryServer.
- EcRgRegistry_1_0
- EcRgRegistryServer_C

library EcDcMsgPng1

/ecs/formal/CSS/DOF/src/MP_OODCE_NO1/msg_oodce_1

This library consists of message passing objects used by both the client and server side. It is used for communication between different servers.

Classes

- EcMpFileInList -- The EcMpFileInList object is the item stored in the persistent incoming message file.
- EcMpFileOutList -- The EcMpFileOutList object is the item stored in the persistent message file for incoming queue (callback or ordinary).
- EcMpMsg -- The EcMpMsg class contains the definitions for message object.
- EcMpMsgCb -- This class is an abstract ecs message callback class.
- EcMpMsgCbList -- Contains information about the callback object.
- EcMpMsgIdList -- Class containing the pair: msg id + receiver log name.

- EcMpMsgPsngCtrl -- The EcMpMsgPsngCtrl object is the controller object, through which any number of receiver sessions can be created.
- EcMpMsgQueue -- This class will be the parent class for the following.
- EcMpMsgQueueCbIn -- FileCleanup Clean the incoming callback persistent file and log in only those message currently in the callback queue EvalThread Get a request from the queue and process the request.
- EcMpMsgQueueIn -- This class will be used to queue the messages once they are received.
- EcMpMsgQueueOut -- In case of a asynchronous calls, the message is put into a queue, which will be sent later by an internal thread.
- EcMpMsgWrapper -- This class is used to wrap up the message and the related information that is common to all the receivers in a session list.
- EcMpQueueInItemList -- The EcMpQueueInItemList object is the item stored in the incoming queue (callback or ordinary queue).
- EcMpQueueInList -- Contains information about the incoming queues.
- EcMpQueueOutItemList -- The EcMpQueueOutItemList object is the item stored in the outgoing queue.
- EcMpReceiverCacheList -- Contains cached information about all the client pointers (sender list).
- EcMpSessionItemList -- List containing information about each receiver in the session list.
- EcMpSessionList -- This class allows to create a list of receivers to which message(s) will be sent.
- EcMpTransfer_1_0 -- This definition declares the interface for this application and associates it with a globally (universally) unique identifier, or UUID.
- EcMpTransfer_1_0_ABS -- Abstract base class of EcMpTransfer.
- EcMpTransfer_1_0_Mgr -- Manager class for EcMpTransfer.
- EcMpTransferCli -- This class is the oodce EcMpTransfer client (surrogate object) It allows you to make requests to an EcMpTransfer manager object, and transfer messages.
- EcMpTransferCliList -- Contains information about the created receiver object.
- EcMpTransferSrv -- This class is the OODCE manager object.

library EcDcMsgPsng1Sh (shared)

/ecs/formal/CSS/DOF/src/MP_OODCE_NO1/msg_oodce_1

This library consists of message passing objects used by both the client and server side.

Classes

- EcMpFileInList -- The EcMpFileInList object is the item stored in the persistent incoming message file.
- EcMpFileOutList -- The EcMpFileOutList object is the item stored in the persistent message file for incoming queue (callback or ordinary).

- EcMpMsg -- The EcMpMsg class contains the definitions for message object.
- EcMpMsgCb -- This class is an abstract ecs message callback class.
- EcMpMsgCbList -- Contains information about the callback object.
- EcMpMsgIdList -- Class containing the pair: msg id + receiver log name.
- EcMpMsgPsngCtrl -- The EcMpMsgPsngCtrl object is the controller object, through which any number of receiver sessions can be created.
- EcMpMsgQueue -- This class will be the parent class for the following.
- EcMpMsgQueueCbIn -- FileCleanup Clean the incoming callback persistent file and log in only those message currently in the callback queue EvalThread Get a request from the queue and process the request.
- EcMpMsgQueueIn -- This class will be used to queue the messages once they are received.
- EcMpMsgQueueOut -- In case of a asynchronous calls, the message is put into a queue, which will be sent later by an internal thread.
- EcMpMsgWrapper -- This class is used to wrap up the message and the related information that is common to all the receivers in a session list.
- EcMpQueueInItemList -- The EcMpQueueInItemList object is the item stored in the incoming queue (callback or ordinary queue).
- EcMpQueueInList -- Contains information about the incoming queues.
- EcMpQueueOutItemList -- The EcMpQueueOutItemList object is the item stored in the outgoing queue.
- EcMpReceiverCacheList -- Contains cached information about all the client pointers (sender list).
- EcMpSessionItemList -- List containing information about each receiver in the session list.
- EcMpSessionList -- This class allows to create a list of receivers to which message(s) will be sent.
- EcMpTransfer_1_0 -- This definition declares the interface for this application and associates it with a globally (universally) unique identifier, or UUID.
- EcMpTransfer_1_0_ABS -- Abstract base class of EcMpTransfer.
- EcMpTransfer_1_0_Mgr -- Manager class for EcMpTransfer.
- EcMpTransferCli -- This class is the oodce EcMpTransfer client (surrogate object) It allows you to make requests to an EcMpTransfer manager object, and transfer messages.
- EcMpTransferCliList -- Contains information about the created receiver object.
- EcMpTransferSrv -- This class is the OODCE manager object.

library EcDmDdClient

/ecs/formal/DM/DDICT/src/ddclient

The library establishes and maintains a session with the DDICT server.

Classes

- DmDdClRequest -- This is the signature of the callback function for asynchronous communication (via the Submit function).
- DmDdClRequestServer -- This class is used by a client to initiate a DDICT session.
- DmDdClSchemaRequest -- This is the signature of the callback function for asynchronous communication (via the Submit function).
- DmDdClUrRequest -- This is the signature of the callback function for asynchronous communication (via the Submit function).

library EcDmLmClient

/ecs/formal/DM/LIMGR/src/client

The EcDmLmClient library contains classes that act as public interfaces to connect to the LIMGR, DIMGR, and ECS to V0 gateway. The interfaces allow the following requests: , granule , browse, and acquire product.

Classes

- DmLmClRequest -- Class contains all methods and attributes related to a user request.
- DmLmClRequestServer -- Class contains all methods and attributes related to tracking user's requests. User can : * instantiate a search request. * suspend session. * resume session. * delete session.

library EcDmLmLimServer

/ecs/formal/DM/LIMGR/src/server

The server CSC provides classes for DM subsystem servers, it contains classes for making LIM server. It makes the LIM server executable program EcDmLmServer. This library is server side library.

Classes

- DmLmDispatcher -- This is the customized version of the dispatcher object.
- DmLmManagedServer -- This class connects to the Process Framework infrastructure during initialization.
- DmLmProcessPlanPair -- This class is used by dispatcher to create DmLmProcessPlan vector list, the vector list in the DmLmDispatcher keeps the DmLmProcessPlan for the reuse.
- DmLmSrRequest -- This class contains all the methods related to a user request that is processed at the server side .

library EcDnDir

/ecs/formal/CSS/DOF/src/DN/directory

This provides classes to manipulate CDS entries, and operations on their attributes using XDS APIs.

Classes

- CsDcXds -- This class provides APIs to DCE-CDS application programming.
- EcDnAttribute -- This class represents an attribute of a CDS entry.
- EcDnCompositeName -- A composite name is a complete CDS path name.
- EcDnContext -- The context class defines the path/set of bindings with distinct atomic names.
- EcDnElement -- The EcDnElement class will contain an element, which is an attribute-value list pair.
- EcDnValue -- This class abstracts the value of an attribute.

library EcDnDirSh (shared)

/ecs/formal/CSS/DOF/src/DN/directory

This provides classes to manipulate CDS entries, and operations on their attributes using XDS APIs.

Classes

- CsDcXds -- This class provides APIs to DCE-CDS application programming.
- EcDnAttribute -- This class represents an attribute of a CDS entry.
- EcDnCompositeName -- A composite name is a complete CDS path name.
- EcDnContext -- The context class defines the path/set of bindings with distinct atomic names.
- EcDnElement -- The EcDnElement class will contain an element, which is an attribute-value list pair.
- EcDnValue -- This class abstracts the value of an attribute.

library EcDpJobMgmtClient

/ecs/formal/PDPS/DPS/PRONG/src/JobMgmtPF

This library provides routines to convert input into a message packet form suitable for communicating with the Job Management Server and then call OODCE routines to initiate and confirm the communication. The format of the messages is specified in the DpPrSchedulerDObj.idl file.

Classes

- DpPrSchedulerDObj_1_0 -- This is a class that is used for communications with other servers.
- DpPrSchedulerProxy -- Class.

library EcFh

/ecs/formal/CSS/DOF/src/FH/src

This library provides fault handling capabilities to ECS applications.

Classes

- EcFhExecutor -- This class invokes the EcFhFaultHandler::HandleDCEOSFException depending on the policy associated with this executor.
- EcFhFaultHandler -- This class knows how to handle exceptions of type DCEOSFException.
- EcFhPolicy -- This class creates the policy for fault handling.

library EcFhSh (shared)

/ecs/formal/CSS/DOF/src/FH/src

This library provides fault handling capabilities to ECS applications.

Classes

- EcFhExecutor -- This class invokes the EcFhFaultHandler::HandleDCEOSFException depending on the policy associated with this executor.
- EcFhFaultHandler -- This class knows how to handle exceptions of type DCEOSFException.
- EcFhPolicy -- This class creates the policy for fault handling.

library EcGwDAR

/ecs/formal/CSS/DOF/src/RELB_GATEWAY/DAR/dar

This library is the DAR Communications client library which client applications should link with to communicate with the DAR Gateway.

Classes

- EcGwDARGetSubxARStatusRequest_C -- This class represents the client side of DARGateway GetSubxARStatus.
- EcGwDARGetxARStatusRequest_C -- This class represents the client side of DARGateway GetxARStatus.
- EcGwDARModifyDarRequest_C -- This class represents the client side of DARGateway ModifyDar.
- EcGwDARQueryxARContentsRequest_C -- This class represents the client side of DARGateway QueryxARContents.
- EcGwDARQueryxARScenesRequest_C -- This class represents the client side of DARGateway QueryxARScenes.
- EcGwDARQueryxARSummaryRequest_C -- This class represents the client side of DARGateway QueryxARSummary.
- EcGwDARSubmitDarRequest_C -- This class represents the client side of DARGateway SubmitDar.
- getSubxARStatusSrfRequest -- This message class, which is derived from the EcUtStreamable class, is used to transport the getSubxARStatus request message by the Server Request Framework (SRF), which is utilized by the ASTER DAR Gateway client library and server for DCE .

- `getSubxARStatusSrfResult` -- This message class, which is derived from the `EcUtStreamable` class, is used to transport the `getSubxARStatus` result message by the Server Request Framework (SRF), which is utilized by the ASTER DAR Gateway client library and server for DCE .
- `getSubxARStatusSrfStatus` -- This message class, which is derived from the `EcUtStreamable` class, is used to transport the `getSubxARStatus` status message by the Server Request Framework (SRF), which is utilized by the ASTER DAR Gateway client library and server for DCE .
- `getxARStatusSrfRequest` -- This message class, which is derived from the `EcUtStreamable` class, is used to transport the `getxARStatus` request message by the Server Request Framework (SRF), which is utilized by the ASTER DAR Gateway client library and server for DCE communication.
- `getxARStatusSrfResult` -- This message class, which is derived from the `EcUtStreamable` class, is used to transport the `getxARStatus` result message by the Server Request Framework (SRF), which is utilized by the ASTER DAR Gateway client library and server for DCE communication.
- `getxARStatusSrfStatus` -- This message class, which is derived from the `EcUtStreamable` class, is used to transport the `getxARStatus` status message by the Server Request Framework (SRF), which is utilized by the ASTER DAR Gateway client library and server for DCE communication.
- `modifyDarSrfRequest` -- This message class, which is derived from the `EcUtStreamable` class, is used to transport the `modifyDar` request message by the Server Request Framework (SRF), which is utilized by the ASTER DAR Gateway client library and server for DCE communication.
- `modifyDarSrfResult` -- This message class, which is derived from the `EcUtStreamable` class, is used to transport the `modifyDar` result message by the Server Request Framework (SRF), which is utilized by the ASTER DAR Gateway client library and server for DCE communication.
- `modifyDarSrfStatus` -- This message class, which is derived from the `EcUtStreamable` class, is used to transport the `modifyDar` status message by the Server Request Framework (SRF), which is utilized by the ASTER DAR Gateway client library and server for DCE communication.
- `queryxARContentsSrfRequest` -- This message class, which is derived from the `EcUtStreamable` class, is used to transport the `queryxARContents` request message by the Server Request Framework (SRF), which is utilized by the ASTER DAR Gateway client library and server for DCE .
- `queryxARContentsSrfResult` -- This message class, which is derived from the `EcUtStreamable` class, is used to transport the `queryxARContents` result message by the Server Request Framework (SRF), which is utilized by the ASTER DAR Gateway client library and server for DCE .
- `queryxARContentsSrfStatus` -- This message class, which is derived from the `EcUtStreamable` class, is used to transport the `queryxARContents` status message by the

Server Request Framework (SRF), which is utilized by the ASTER DAR Gateway client library and server for DCE .

- queryxARScenesSrfRequest -- This message class, which is derived from the EcUtStreamable class, is used to transport the queryxARScenes request message by the Server Request Framework (SRF), which is utilized by the ASTER DAR Gateway client library and server for DCE .
- queryxARScenesSrfResult -- This message class, which is derived from the EcUtStreamable class, is used to transport the queryxARScenes result message by the Server Request Framework (SRF), which is utilized by the ASTER DAR Gateway client library and server for DCE .
- queryxARScenesSrfStatus -- This message class, which is derived from the EcUtStreamable class, is used to transport the queryxARScenes status message by the Server Request Framework (SRF), which is utilized by the ASTER DAR Gateway client library and server for DCE .
- queryxARSummarySrfRequest -- This message class, which is derived from the EcUtStreamable class, is used to transport the queryxARSummary request message by the Server Request Framework (SRF), which is utilized by the ASTER DAR Gateway client library and server for DCE .
- queryxARSummarySrfResult -- This message class, which is derived from the EcUtStreamable class, is used to transport the queryxARSummary result message by the Server Request Framework (SRF), which is utilized by the ASTER DAR Gateway client library and server for DCE .
- queryxARSummarySrfStatus -- This message class, which is derived from the EcUtStreamable class, is used to transport the queryxARSummary status message by the Server Request Framework (SRF), which is utilized by the ASTER DAR Gateway client library and server for DCE .
- submitDarSrfRequest -- This message class, which is derived from the EcUtStreamable class, is used to transport the submitDar request message by the Server Request Framework (SRF), which is utilized by the ASTER DAR Gateway client library and server for DCE communication.
- submitDarSrfResult -- This message class, which is derived from the EcUtStreamable class, is used to transport the submitDar result message by the Server Request Framework (SRF), which is utilized by the ASTER DAR Gateway client library and server for DCE communication.
- submitDarSrfStatus -- This message class, which is derived from the EcUtStreamable class, is used to transport the submitDar status message by the Server Request Framework (SRF), which is utilized by the ASTER DAR Gateway client library and server for DCE communication.

library EcHtSh (shared)

/ecs/formal/IOS/AdvService/src/EcHtmlLib

This library provides classes that contain operations to handle all HTML tags. Primarily, its functions are used to create dynamic HTML pages through CGI.

Classes

- EcHtAddress -- The EcHtAddress class provides the user interface to the representation holding the object's state.
- EcHtAddressRep -- Class which represents the HTML Address element.
- EcHtBaseHandle -- This class provides an abstract base class for a reference counting scheme.
- EcHtCgiElement -- Class to parse arguments from a Web server for use by a cgi-bin program.
- EcHtCgiElementKey -- Class to parse arguments from a Web server for use by a cgi-bin program.
- EcHtCgiVars -- Class to parse arguments from a Web server for use by a cgi-bin program.
- EcHtCode -- The EcHtCode class provides the user interface to the representation holding the object's state.
- EcHtCodeRep -- Class representing the HTML CODE element intended for short words or phrases of code; the PRE block structuring element is more appropriate for multiple-line listings.
- EcHtCustomList -- The EcHtCustomList class provides the user interface to the representation holding the object's state.
- EcHtCustomListRep -- This class provides an interface to the HTML unordered list construct.
- EcHtDefinitionListRep -- Class representing the HTML Defintion List (<DL>, <DT>, <DD>, and </DL>), a list of terms and corresponding defintions.
- EcHtDirectoryList -- The EcHtDirectoryList class provides the user interface to the representation holding the object's state.
- EcHtDirectoryListRep -- Class to represent the HTML Directory List (, , and elements).
- EcHtElement -- The handle class provides the client interface to the representation that holds the object's state. Object instances of this class should be declared normally, but operations should always be through the "operator->". This operator forwards operations .
- EcHtElementRep -- This is the abstract base class for all HTML construct representation classes.
- EcHtForm -- Class to represent an HTML Form.
- EcHtFormCheckbox -- Class to represent a Form Check Box Input Element.
- EcHtFormCheckboxRep -- Class to represent a Form Check Box Input Element.
- EcHtFormHidden -- Class to represent a Form Hidden Input Element.
- EcHtFormHiddenRep -- Class to represent a Form Hidden Input Element.
- EcHtFormImage -- Class to represent a Form Image Input Element.
- EcHtFormImageRep -- Class to represent a Form Image Input Element.
- EcHtFormPassword -- Class to represent a Form Passord Input Element.

- EcHtFormPasswordRep -- Class to represent a Form Password Input Element.
- EcHtFormRadio -- Class to represent a Form Radio Input Element.
- EcHtFormRadioRep -- Class to represent a Form Radio Input Element.
- EcHtFormRep -- Class to represent an HTML Form.
- EcHtFormReset -- The EcHtFormReset class provides the user interface to the EcHtFormResetRep concrete representation class.
- EcHtFormResetRep -- Class to represent a Form Reset Input Element.
- EcHtFormSelect -- The EcHtFormSelect class provides the user interface to the EcHtFormSelectRep concrete representation class.
- EcHtFormSelectRep -- Class to represent a Form Select Element.
- EcHtFormSubmit -- The EcHtFormSubmit class provides the user interface to the EcHtFormSubmitRep concrete representation class.
- EcHtFormSubmitRep -- Class to represent a Form Submit Input Element.
- EcHtFormText -- The EcHtFormText class provides the user interface to the EcHtFormTextRep concrete representation class.
- EcHtFormTextArea -- Class to represent an HTML Form Text Area Element.
- EcHtFormTextAreaRep -- The EcHtFormTextArea class provides the user interface to the EcHtFormTextAreaRep concrete representation class.
- EcHtFormTextRep -- Class to represent an HTML Form Text Input Element.
- EcHtHeader -- The EcHtHeader class provides the user interface to the EcHtHeaderRep concrete representation class.
- EcHtHeaderRep -- Class representing HTML <H>, </H> element for section headings.
- EcHtImageRep -- Class for an HTML IMG element for referring to an image or icon via a hyperlink.
- EcHtLinkRep -- Class to represent a hyperlink.
- EcHtListRep -- This is the abstract base class for all HTML lists.
- EcHtOrderedList -- User interface to the EcHtOrderedListRep class.
- EcHtOrderedListRep -- Class to represent the HTML Ordered List (, , and elements) for an ordered list of items (typically rendered as a numbered list).
- EcHtPageRep -- Class representing HTML page consisting of header, body and footer.
- EcHtParagraph -- User interface to the EcHtParagraphRep class.
- EcHtParagraphRep -- Class for representing an HTML <P> element for paragraphs.
- EcHtPreformattedText -- User interface to the EcHtPreformattedText class.
- EcHtPreformattedTextRep -- Class representing the HTML Preformatted Text (<PRE> and </PRE> elements) for a character cell block of text and is suitable for text that has been formatted for a monospaced font.

- EcHtRegion -- User interface to the EcHtRegionRep class.
- EcHtRegionRep -- This class is a concrete base class to be used by callers/derived representations that contain many HTML elements.
- EcHtTable -- Provides the representation of the HTML Table tag and its attributes.
- EcHtTableElements -- Provides the representation of the HTML Table elements and their attributes.
- EcHtUnorderedList -- User interface to the EcHtUnorderedListRep class.
- EcHtUnorderedListRep -- Class representing the HTML Unordered List (, , and elements) for a list of items (typically rendered as a bulleted list).

library EcMsAgDeputyClient

/ecs/formal/MSS/MACI/src

Classes

- MsAgDeputyOD_1_0

library EcNsServiceLoc

/ecs/formal/CSS/DOF/src/NS/naming

This library contains both the client and server side APIs to allow servers to advertise their services and to allow clients to locate the servers. The primary server side functions are ExportBinding and GroupMemberAdd. These functions allow a server to advertise its services in CDS. The primary client side function is GetServerBinding which is responsible for querying CDS for a desired service and returning a valid binding to the application.

Classes

- EcNsServiceLoc -- This class captures the common operations and state used by the client and server derivatives of this class.
- EcNsServiceLocClient -- The class supports multiple constructors to accomodate various means of specifying a service.
- EcNsServiceLocServer -- This class constructs a CDS path for a server, allows that path to be added to an RPC group and allows both the RPC group and the server path to be exported to CDS.

library EcNsServiceLocator

/ecs/formal/CSS/DOF/src/NS/service_locator

The Service Locator library provides an implementation independent mechanism for specifying service locations. The mechanism is called the service token and essentially consists of a cell name and a service name. There are two parts to the library. The first is the service locator class which encapsulates the service token management APIs. The second is the locator mapper class which provides access to a service locator mapping file. The mapping file maps service/cell long names to shortened versions of the name. For instance, the cell

/.../gsfcell.gsfc.nasa.gov is mapped to GSF and the service name EcSbSubServer is mapped to CSSBSRV.

Classes

- EcNsLocatorMapper -- This class maps long CDS entries like cell names and server group entries to and from short representations.
- EcNsServiceLocator -- This Class provide technology independant means of describing an ECS service.

library EcNsServiceLocatorSh (shared)

/ecs/formal/CSS/DOF/src/NS/service_locator

The Service Locator library provides an implementation independent mechanism for specifying service locations. The mechanism is called the service token and essentially consists of a cell name and a service name. There are two parts to the library. The first is the service locator class which encapsulates the service token management APIs. The second is the locator mapper class which provides access to a service locator mapping file. The mapping file maps service/cell long names to shortened versions of the name. For instance, the cell /.../gsfcell.gsfc.nasa.gov is mapped to GSF and the service name EcSbSubServer is mapped to CSSBSRV.

Classes

- EcNsLocatorMapper -- This class maps long CDS entries like cell names and server group entries to and from short representations.
- EcNsServiceLocator -- This Class provide technology independant means of describing an ECS service.

library EcNsServiceLocSh (shared)

/ecs/formal/CSS/DOF/src/NS/naming

This library contains both the client and server side APIs to allow servers to advertise their services and to allow clients to locate the servers. The primary server side functions are ExportBinding and GroupMemberAdd. These functions allow a server to advertise its services in CDS. The primary client side function is GetServerBinding which is responsible for querying CDS for a desired service and returning a valid binding to the application.

Classes

- EcNsServiceLoc -- This class captures the common operations and state used by the client and server derivatives of this class.
- EcNsServiceLocClient -- The class supports multiple constructors to accomodate various means of specifying a service.
- EcNsServiceLocServer -- This class constructs a CDS path for a server, allows that path to be added to an RPC group and allows both the RPC group and the server path to exported to CDS.

library EcPf

/ecs/formal/CSS/DOF/src/PF/pf

This library provides flexible mechanism for ECS client and server applications to transparently include ECS infrastructure features from a library of services.

Classes

- EcPfClient -- Class that provides wrappers for OODCE and DCE functionality.
- EcPfGenProcess -- EcPfGenProcess is an abstract class which represents the process framework for a generic process.
- EcPfGenServer -- The class will provide the generic DCEServer capabilities and some enhanced features as well.
- EcPfManagedServer -- This class provides the functionality of hiding the interface between the MSS and the application by instantiating the EcAgManger object and making calls to it for the application, and it also provides an interface for the MSS to call on the application.
- EcPfProxy -- Currently does not maintain any information or provide any functionality.
- EcPfUnmanagedServer -- ointer to a PfClient, In otherwords it is a pointer to itself.
- ObjectLinkList -- This class maintains a linked list of structures ObjectLink whose fields are DCEObject and a pointer to the next structure in the list.

library EcPfSh (shared)

/ecs/formal/CSS/DOF/src/PF/pf

This library provides flexible mechanism for ECS client and server applications to transparently include ECS infrastructure features from a library of services.

Classes

- EcPfClient -- Class that provides wrappers for OODCE and DCE functionality.
- EcPfGenProcess -- EcPfGenProcess is an abstract class which represents the process framework for a generic process.
- EcPfGenServer -- The class will provide the generic DCEServer capabilities and some enhanced features as well.
- EcPfManagedServer -- This class provides the functionality of hiding the interface between the MSS and the application by instantiating the EcAgManger object and making calls to it for the application, and it also provides an interface for the MSS to call on the application.
- EcPfProxy -- Currently does not maintain any information or provide any functionality.
- EcPfUnmanagedServer -- ointer to a PfClient, In otherwords it is a pointer to itself.
- ObjectLinkList -- This class maintains a linked list of structures ObjectLink whose fields are DCEObject and a pointer to the next structure in the list.

library EcPo

/ecs/formal/IOS/AdvService/src/PersistentLib

The Persistent CSC provides a set of abstract and base classes for approved advertisements and requests. It also provides classes which support caching and defer fetching mechanism.

Classes

- EcPoCache -- This class supports a cache of different different subclasses of EcPoIdentifiedRepresentation.
- EcPoCachingHandle -- This class works in concert with EcPoIdentifiedRepresentation to support a cache of representation in addition to acting as a EcPoIdentifiedHandle.
- EcPoCoordFetch -- This class is the abstract base class for all objects that are RWDB Tools to do their persistence and allow other objects to coordinate thier fetches with it to allow for multiple objects to be fetched form the database with a single hit.
- EcPoDependant -- If a object contains another object it is typical for the life of the sub-object to be constrained by the super-object.
- EcPoHandle -- This class works in concert with EcPoRepresentation as abstract base classes for a persistent object framework.
- EcPoIdentifiedHandle -- This class works in concert with EcPoIdentifiedRepresentation to provide a convenient deferred fetching handle with an integer ID.
- EcPoIdentifiedRepresentation -- We are a specialization of a EcPoRepresentation with the concept of a defined primary key, our IdentityState.
- EcPoIdentityState
- EcPoLongString -- This class is a RWCString that can insert itself into the database.
- EcPoPersistent -- Declares a class, EcPoPersistent, to act as the abstract base class for all persistent objects.
- EcPoRepresentation -- This class is the abstract base class for all objects that are persistent and are part of the deferred fetching scheme.
- EcPoUpdateAware -- This class is a specialization of the EcPoCoordFetch class.

library EcPoDbRW

/ecs/formal/COMMON/CSCI_DBWrapper/src

This library provides database connection wrappers for Rogue Wave DBtools++ library with additional connection pooling, error checking and tolerance features for PF clients.

Classes

- EcPoConnectionsRW -- This class provides access to specific global objects by providing the handle for them.
- EcPoDb -- Objects of this class represent an RWDB Database together with the connections in use by the database.
- EcPoError -- This class provides a series of misc.

- EcPoStrategy -- Objects of this class represent error recovery strategies.
- EcPoThread -- Objects of this class represent a single RWDB connection and the thread ID of the thread that claimed the connection.

library EcPoDbRWSh (shared)

/ecs/formal/COMMON/CSCI_DBWrapper/src

Contains methods that allow access to a RogueWave database together with the current list of connections to the database. Uses RogueWave APIs.

Classes

- EcPoConnectionsRW -- This class provides access to specific global objects by providing the handle for them.
- EcPoDb -- Objects of this class represent an RWDB Database together with the connections in use by the database.
- EcPoError -- This class provides a series of misc.
- EcPoStrategy -- Objects of this class represent error recovery strategies.
- EcPoThread -- Objects of this class represent a single RWDB connection and the thread ID of the thread that claimed the connection.

library EcPoSh (shared)

/ecs/formal/IOS/AdvService/src/PersistentLib

The Persistent CSC provides a set of abstract and base classes for approved advertisements and requests. It also provides classes which support caching and defer fetching mechanism.

Classes

- EcPoCache -- This class supports a cache of different different subclasses of EcPoIdentifiedRepresentation.
- EcPoCachingHandle -- This class works in concert with EcPoIdentifiedRepresentation to support a cache of representation in addition to acting as a EcPoIdentifiedHandle.
- EcPoCoordFetch -- This class is the abstract base class for all objects that are RWDB Tools to do their persistence and allow other objects to coordinate thier fetches with it to allow for multiple objects to be fetched form the database with a single hit.
- EcPoDependant -- If a object contains another object it is typical for the life of the sub-object to be constrained by the super-object.
- EcPoHandle -- This class works in concert with EcPoRepresentation as abstract base classes for a persistent object framework.
- EcPoIdentifiedHandle -- This class works in concert with EcPoIdentifiedRepresentation to provide a convenient deferred fetching handle with an integer ID.
- EcPoIdentifiedRepresentation -- We are a specialization of a EcPoRepresentation with the concept of a defined primary key, our IdentityState.

- EcPoIdentityState
- EcPoLongString -- This class is a RWCString that can insert itself into the database.
- EcPoPersistent -- Declares a class, EcPoPersistent, to act as the abstract base class for all persistent objects.
- EcPoRepresentation -- This class is the abstract base class for all objects that are persistent and are part of the deferred fetching scheme.
- EcPoUpdateAware -- This class is a specialization of the EcPoCoordFetch class.

library EcPtSpecialLock

/ecs/formal/CSS/DOF/src/THREAD/locks

This library provides classes for mutex operations and thread IDs.

Classes

- EcPtSharedLock -- EcPtSharedLock is a class of shared/exclusive mutexes.
- EcPtThreadId -- This is an ID for a thread.
- EcPtUnlockingMutex -- EcPtUnlockingMutex is a class of self unlocking mutexes.

library EcPtSpecialLockSh (shared)

/ecs/formal/CSS/DOF/src/THREAD/locks

This library provides classes for mutex operations and thread IDs.

Classes

- EcPtSharedLock -- EcPtSharedLock is a class of shared/exclusive mutexes.
- EcPtThreadId -- This is an ID for a thread.
- EcPtUnlockingMutex -- EcPtUnlockingMutex is a class of self unlocking mutexes.

library ECSAuthn

/ecs/formal/COMMON/CSCI_Util/src/EcsAuthenticator

This library provides a function (userID, password, ecsAuth), which converts the input of a userID and a password of this userID, and returns a string contains the encrypted authenticator ecsAuth.

library EcSbCI

/ecs/formal/CSS/DOF/src/SUBSCRIPTION/cl

This library contains all the classes for Subscription Server client.

Classes

- EcCICancelAllSubscriptionsCb -- Class for obtaining subscription data.
- EcCICancelEventIDSubscriptionsCb -- Client callback class for cancelling subscriptions with an event ID.

- `EcClCancelExpDateSubscriptionsCb` -- Client callback class for cancelling subscriptions with an expiration date.
- `EcClCancelSubCb`
- `EcClCancelUserIDSubscriptionsCb` -- Class for obtaining subscription data.
- `EcClEvent` -- Event class represented on the client side.
- `EcClEventCollector` -- This class provides a collection mechanism for retrieving and manipulating multiple events.
- `EcClGetAllEventsCb` -- Class for obtaining subscription data.
- `EcClGetAllSubscriptionsCb` -- Class for obtaining subscription data.
- `EcClGetCatEventsCb` -- Class for obtaining subscription data.
- `EcClGetCreatorEventsCb` -- Class for obtaining subscription data.
- `EcClGetEventDataCb` -- Client callback class for obtaining event data.
- `EcClGetEventIDSubscriptionsCb` -- Class for obtaining subscription data.
- `EcClGetExpDateSubscriptionsCb` -- Class for obtaining subscription data.
- `EcClGetSubDataCb` -- Class for obtaining subscription data.
- `EcClGetUserIDSubscriptionsCb` -- Class for obtaining subscription data.
- `EcClRegisterEventCb` -- Callback class for registering events.
- `EcClRestartNotifyCb` -- Callback class for Subscription Server restart notification.
- `EcClSubmitSubCb` -- Callback class for submit a new subscription.
- `EcClSubscription` -- Client side subscription.
- `EcClSubscriptionCollector` -- This class provides a collection mechanism for retrieving and manipulating multiple subscriptions.
- `EcClTriggerEventCb` -- Callback class for triggering an event.
- `EcClUnregisterAllEventsCb` -- Class for obtaining subscription data.
- `EcClUnregisterCatEventsCb` -- Class for obtaining subscription data.
- `EcClUnregisterCreatorEventsCb` -- Class for obtaining subscription data.
- `EcClUnregisterEventCb` -- Callback class for unregistering an event.
- `EcClUpdateEventCb` -- Callback class for update an event.
- `EcClUpdateSubCb` -- Callback class for updating a subscription.

library EcSbMsg

`/ecs/formal/CSS/DOF/src/SUBSCRIPTION/msg`

This library contains all message classes used for the communication between Subscription Server and its client objects provided by Subscription Server client library. It is used by Subscription Server and its client only.

Classes

- `EcSbAcquireAction` -- This is a subclass of `EcSbGenAction` which is derived from `EcUtStreamable`.
- `EcSbEventUR` -- The `EcSbEventUR` class is derived from `EcUrUR` to provide an `EventUR` type class for Event server to use.
- `EcSbEventURProvider` -- Data entity enables this class to quickly retrieve the object parameters.
- `EcSbGenAction` -- All the actions specified by subscriber should subclass from it.
- `EcSbNotification` -- This class embeds all notification information which server uses to notify user when action is taken place.
- `EcSbSubUR`
- `EcSbSubURProvider` -- Subscription UR provider class.
- `MsgCancelAllSubscriptions` -- Message class for cancelling all subscriptions.
- `MsgCancelEvent` -- Message class for cancelling a event.
- `MsgCancelEventIDSubscriptions` -- Message class for cancelling subscriptions for a given event ID.
- `MsgCancelExpDateSubscriptions` -- Class for cancelling subscriptions by their expiration date.
- `MsgCancelSubscription` -- Message class for cancelling a subscription.
- `MsgCancelUserIDSubscriptions` -- Message class for cancelling subscriptions by their user ID.
- `MsgGenResult` -- Class generated automatically by the Message Class Generator.
- `MsgGenStatus` -- Message class automatically generated using the Message Class Generator.
- `MsgGetAllEvents` -- Message class for retrieving all events.
- `MsgGetAllSubscriptions` -- Message class for retrieving all subscriptions.
- `MsgGetCatEvents` -- Message class for retrieving events by their category.
- `MsgGetCreatorEvents` -- Message class for retrieving events by their creator.
- `MsgGetEventData` -- The message class for retrieving event data.
- `MsgGetEventDataResult` -- Message class to retrieve event data.
- `MsgGetEventIDSubscriptions` -- Message class used for retrieving subscriptions for a given event ID.
- `MsgGetExpDateSubscriptions` -- Message class for retrieving subscriptions by the expiration date.
- `MsgGetFullSubDataResult`
- `MsgGetSubDataResult` -- Message class for retrieving subscription data.
- `MsgGetSubscriptionData` -- Message class for retrieving a subscription.
- `MsgGetUserIDSubscriptions` -- Message class for retrieving subscriptions by user ID.

- `MsgListResult` -- Message class for retrieving a list of objects.
- `MsgRegisterEvent` -- Message class for information on registering an event.
- `MsgRegisterEventResult` -- Message class for registering an event's result.
- `MsgRestartNotify` -- Message class for restart notification.
- `MsgSubmitSubResult` -- Message class for returning the result of submit a subscription.
- `MsgSubmitSubscription` -- Message class for submitting a subscription.
- `MsgTriggerEvent` -- Message class for triggering an event.
- `MsgUnregisterAllEvents` -- Message class for unregistering all events.
- `MsgUnregisterCatEvents` -- Message class for unregistering events by category.
- `MsgUnregisterCreatorEvents` -- Message class for unregistering events by their creator.
- `MsgUpdateEvent` -- Message class for updating an event.
- `MsgUpdateSubscription` -- Message class for updating a subscription.

library EcSbSr

/ecs/formal/CSS/DOF/src/SUBSCRIPTION/sb

This library consists of all subscription server side class objects. It is used by `EcSbSubServer` program only.

Classes

- `EcSbCancelAllSubsRequest` -- This is the server side object.
- `EcSbCancelEventIDSubsRequest` -- Handles a request of cancelling all subscription associated with a given event by the server.
- `EcSbCancelEventRequest` -- Handles a request of cancelling a specified event by the server.
- `EcSbCancelExpDateSubsRequest` -- Handles a request of cancelling all subscription whose expiration date is beyond specified date.
- `EcSbCancelSubRequest` -- Handles a request of cancelling a specified subscription by the server.
- `EcSbCancelUserIDSubsRequest` -- Handles a request of cancelling all subscription subscribed by a specified user.
- `EcSbDbInterface` -- This class provides utilities to access to database and the interface to execute sql and stored procedures.
- `EcSbEvent` -- This is the server side event representation, customized to implement the actual operations supported by the server.
- `EcSbEventDispatcher` -- This is the customized version of the dispatcher object.
- `EcSbEventHandler` -- This class provides the interface to work with multiple events (i.e., aggregate behaviour) on the server side.
- `EcSbEventRServer` -- This is a wrapper class to wrap up `EcSrRequestServer_S` for the Event Server Object.

- EcSbEventStore -- This class provides the means for adding events to the persistent store, instantiating events from the persistent store, and passing the events to the subscription server via EcSbEventHandler.
- EcSbGetAllEventsRequest -- Handles a request of getting all events by the server.
- EcSbGetAllSubsRequest -- Handles a request of getting all subscription by the server.
- EcSbGetCatEventsRequest -- Handles a request of getting an event by given event category.
- EcSbGetCreatorEventsRequest -- Handles a request of getting all events created by a given user.
- EcSbGetEventIDSubsRequest -- Handles a request of getting all subscription associated with given event id.
- EcSbGetEventRequest -- Handles a request of getting all events from server.
- EcSbGetExpDateSubsRequest -- Handles a request of getting all subscription by given expiration data.
- EcSbGetSubRequest -- Handles a request of getting all subscription by the server.
- EcSbGetUserIDSubsRequest -- Handles a request of retrieve subscription subscribed by a given user.
- EcSbManagedServer
- EcSbPersistWorkThreads -- This class keeps track of EcSbPersistWorkThreads table.
- EcSbRegisterEventRequest -- This class represent an event register request in subscription server.
- EcSbRestartNotifyRequest
- EcSbSubmitSubRequest -- Handles a request of submitting a subscription a client.
- EcSbSubscription
- EcSbSubscriptionDispatcher
- EcSbSubscriptionHandler -- This class provides the mechanism for collecting and passing around EcSbSubscriptions.
- EcSbSubscriptionRServer -- This is the server wrapper class.
- EcSbSubscriptionStore -- This class provides the means for adding subscriptions to the persistent store, instantiating subscriptions from the persistent store, and passing the subscriptions to the subscription server via EcSbSubscriptionHandler.
- EcSbTimeKeeper -- This class keeps track of time as the subscription server runs and makes sure that anything linked to time (such as the expiration of subscriptions) occurs.
- EcSbTriggerEventRequest -- Handles a request of triggering an event by the server.
- EcSbTriggerRequest
- EcSbUnregisterAllEventsRequest -- Handles a request of cancelling all events by the server.
- EcSbUnregisterCatEventsRequest -- Handles a request of cancelling all events related to specified category.

- EcSbUnregisterCreatorEventsRequest -- Handle a request of cancelling all event related to a specified user.
- EcSbUpdateEventRequest -- Handles a request of updating an event by the server.
- EcSbUpdateSubRequest -- Handles a request of updating a subscription by server.

library EcSbSrSh (shared)

/ecs/formal/CSS/DOF/src/SUBSCRIPTION/sb

This library is consist of all subscription server side class objects. It is used by EcSbSubServer program only.

Classes

- EcSbCancelAllSubsRequest -- This is the server side object.
- EcSbCancelEventIDSubsRequest -- Handles a request of cancelling all subscription associated with a given event by the server.
- EcSbCancelEventRequest -- Handles a request of cancelling a specified event by the server.
- EcSbCancelExpDateSubsRequest -- Handles a request of cancelling all subscription whose expiration date is beyond specified date.
- EcSbCancelSubRequest -- Handles a request of cancelling a specified subscription by the server.
- EcSbCancelUserIDSubsRequest -- Handles a request of cancelling all subscription subscribed by a specified user.
- EcSbDbInterface -- This class provides utilities to access to database and the interface to execute sql and stored procedures.
- EcSbEvent -- This is the server side event representation, customized to implement the actual operations supported by the server.
- EcSbEventDispatcher -- This is the customized version of the dispatcher object.
- EcSbEventHandler -- This class provides the interface to work with multiple events (i.e., aggregate behaviour) on the server side.
- EcSbEventRServer -- This is a wrapper class to wrap up EcSrRequestServer_S for the Event Server Object.
- EcSbEventStore -- This class provides the means for adding events to the persistent store, instantiating events from the persistent store, and passing the events to the subscription server via EcSbEventHandler.
- EcSbGetAllEventsRequest -- Handles a request of getting all events by the server.
- EcSbGetAllSubsRequest -- Handles a request of getting all subscription by the server.
- EcSbGetCatEventsRequest -- Handles a request of getting an event by given event category.
- EcSbGetCreatorEventsRequest -- Handles a request of getting all events created by a given user.
- EcSbGetEventIDSubsRequest -- Handles a request of getting all subscription associated with given event id.

- EcSbGetEventRequest -- Handles a request of getting all events from server.
- EcSbGetExpDateSubsRequest -- Handles a request of getting all subscription by given expiration data.
- EcSbGetSubRequest -- Handles a request of getting all subscription by the server.
- EcSbGetUserIDSubsRequest -- Handles a request of retrieve subscription subscribed by a given user.
- EcSbManagedServer
- EcSbPersistWorkThreads -- This class keeps track of EcSbPersistWorkThreads table.
- EcSbRegisterEventRequest -- This class represent an event register request in subscription server.
- EcSbRestartNotifyRequest
- EcSbSubmitSubRequest -- Handles a request of submitting a subscription a client.
- EcSbSubscription
- EcSbSubscriptionDispatcher
- EcSbSubscriptionHandler -- This class provides the mechanism for collecting and passing around EcSbSubscriptions.
- EcSbSubscriptionRServer -- This is the server wrapper class.
- EcSbSubscriptionStore -- This class provides the means for adding subscriptions to the persistent store, instantiating subscriptions from the persistent store, and passing the subscriptions to the subscription server via EcSbSubscriptionHandler.
- EcSbTimeKeeper -- This class keeps track of time as the subscription server runs and makes sure that anything linked to time (such as the expiration of subscriptions) occurs.
- EcSbTriggerEventRequest -- Handles a request of triggering an event by the server.
- EcSbTriggerRequest
- EcSbUnregisterAllEventsRequest -- Handles a request of cancelling all events by the server.
- EcSbUnregisterCatEventsRequest -- Handles a request of cancelling all events related to specified category.
- EcSbUnregisterCreatorEventsRequest -- Handlea a request of cancelling all event related to a specified user.
- EcSbUpdateEventRequest -- Handles a request of updating an event by the server.
- EcSbUpdateSubRequest -- Handles a request of updating a subscription by server.

library EcSeCmi

/ecs/formal/CSS/DOF/src/AUTHN/authn

Cryptographic Management Interface (CMI) provides a means for processes to obtain randomized passwords with which to gain access to non-DCE services (e.g., Sybase). This library generates a randomized username and password based on a given seed. The seed is combined with a data file of highly randomized data (EcSeRandomDataFile) and uses an indexing algorithm to obtain a randomized username and password.

Classes

- EcSeCmi -- This class will generate a user ID and password based on a given application key.

library EcSeCmiSh (shared)

/ecs/formal/CSS/DOF/src/AUTHN/authn

Cryptographic Management Interface (CMI) provides a means for processes to obtain randomized passwords with which to gain access to non-DCE services (e.g., Sybase). This library generates a randomized username and password based on a given seed. The seed is combined with a data file of highly randomized data (EcSeRandomDataFile) and uses an indexing algorithm to obtain a randomized username and password.

Classes

- EcSeCmi -- This class will generate a user ID and password based on a given application key.

library EcSeLogin

/ecs/formal/CSS/DOF/src/SEC/login/loginlib

This library creates a DCE login context for a process. It differs from the EcSeAuthn and EcSeServerKeyMgmt libraries in that it creates the login context by fork/exec'ing the dce_login process. This library is only primarily used by the EcSeLoginProg DCE login utility.

library EcSeLoginContext

/ecs/formal/CSS/DOF/src/SEC/logincontext

This library uses the key obtained from either the EcSeServerKeyMgmt or EcSeAuthn library to establish a login context for a process.

Classes

- EcSeLoginContext -- Any file that uses EcSeLoginContext must include this header file.

library EcSeLoginContextSh (shared)

/ecs/formal/CSS/DOF/src/SEC/logincontext

This library uses the key obtained from either the EcSeServerKeyMgmt or EcSeAuthn library to establish a login context for a process.

Classes

- EcSeLoginContext -- Any file that uses EcSeLoginContext must include this header file.

library EcSeServerKeyMgmt

/ecs/formal/CSS/DOF/src/SEC/keymgmt

This library provides a mechanism for servers to read their DCE login key from a keytab file. This is the most used method by which servers login to DCE. Keytab files are stored in the \$ECS_HOME/CUSTOM/security directory.

Classes

- EcSeServerKeyMgmt -- The EcSeServerKeyMgmt class is the concrete implementation of the abstract DCEPassword class.

library EcSeServerKeyMgmtSh (shared)

/ecs/formal/CSS/DOF/src/SEC/keymgmt

This library provides a mechanism for servers to read their DCE login key from a keytab file. This is the most used method by which servers login to DCE. Keytab files are stored in the \$ECS_HOME/CUSTOM/security directory.

Classes

- EcSeServerKeyMgmt -- The EcSeServerKeyMgmt class is the concrete implementation of the abstract DCEPassword class.

library EcSeSybSecurityStubbed

/ecs/formal/CSS/DOF/src/SEC/sybsecuritystubbs

This library is the stubbed version of . Used only for completing the successful compilation.. When the applictaion needed the security, it has to replaced by the acutal library.

Classes

- AclInfo -- This class is the one which records all the acl information such as dbname, aclname, principle name, print type, and permission.
- ECS_sec_id_pac_t -- This class contains methods to write and read the sec_id_pac_t into and from a memory stream.
- EcSeGenSecurity -- Any file that use the EcSeGenSecurity class, its methods or its global variable EcXSeSecurity must include this header file.
- EcSeSybSecurity -- Any file that use the EcSeSybSecurity class, its methods or its global variable EcXSeSecurity must include this header file.
- SchemaInfo -- This class has to do with ACL database, which has been deprioritized.

library EcSeSybSecurityStubbedSh (shared)

/ecs/formal/CSS/DOF/src/SEC/sybsecuritystubbs

This library is the stubbed version of . Used only for completing the successful compilation.. When the applictaion needed the security, it has to replaced by the acutal library.

Classes

- AclInfo -- This class is the one which records all the acl information such as dbname, aclname, principle name, print type, and permission.
- ECS_sec_id_pac_t -- This class contains methods to write and read the sec_id_pac_t into and from a memory stream.
- EcSeGenSecurity -- Any file that use the EcSeGenSecurity class, its methods or its global variable EcXSeSecurity must include this header file.
- EcSeSybSecurity -- Any file that use the EcSeSybSecurity class, its methods or its global variable EcXSeSecurity must include this header file.
- SchemaInfo -- This class has to do with ACL database, which has been deprioritized.

library EcSeUtilityCtStubbed

/ecs/formal/CSS/DOF/src/SEC/sybsecuritystubbs

This library is stub version of the EcSeUtilityCt library. Used only for successful compilation of the Application.

Classes

- EcDbInterface -- The following functions allow users to connect to database, execute SQL statements, verify connections states, and disconnect from the database.
- ECS_sec_id_pac_t -- This class contains methods to write and read the sec_id_pac_t into and from a memory stream.
- FetchAclInfo -- This Class is for Sybase database ineraction.

library EcSeUtilityCtStubbedSh (shared)

/ecs/formal/CSS/DOF/src/SEC/sybsecuritystubbs

This library is stub version of the EcSeUtilityCt library. Used only for successful compilation of the Application.

Classes

- EcDbInterface -- The following functions allow users to connect to database, execute SQL statements, verify connections states, and disconnect from the database.
- ECS_sec_id_pac_t -- This class contains methods to write and read the sec_id_pac_t into and from a memory stream.
- FetchAclInfo -- This Class is for Sybase database ineraction.

library EcSeUtilityDbStubbed

/ecs/formal/CSS/DOF/src/SEC/sybsecuritystubbs

This library is stub version of the EcSeUtilityDb library. Used only for successful compilation of the Application.

Classes

- ECS_sec_id_pac_t -- This class contains methods to write and read the sec_id_pac_t into and from a memory stream.
- FetchAclInfo -- This Class is for Sybase database ineraction.

library EcSeUtilityDbStubbedSh (shared)

/ecs/formal/CSS/DOF/src/SEC/sybsecuritystubbs

This library is stub version of the EcSeUtilityDb library. Used only for successful compilation of the Application.

Classes

- ECS_sec_id_pac_t -- This class contains methods to write and read the sec_id_pac_t into and from a memory stream.
- FetchAclInfo -- This Class is for Sybase database ineraction.

library EcTiTime

/ecs/formal/CSS/DOF/src/TIME/time

This library is used to obtain the current time in various formats, and manipulate the time. There will be no capability for the programmer to set the time.

Classes

- EcTiTimeService -- This class is used to obtain the current time in various formats, and manipulate the time.

library EcTiTimeSh (shared)

/ecs/formal/CSS/DOF/src/TIME/time

This library is used to obtain the current time in various formats, and manipulate the time. There will be no capability for the programmer to set the time.

Classes

- EcTiTimeService -- This class is used to obtain the current time in various formats, and manipulate the time.

library EcUr

/ecs/formal/COMMON/CSCI_UR/src/UR/framework

A Universal Reference (UR) is created as a logical reference to a extant ECS object. This reference can be passed around, written to disk, read from disk, etc. A UR is used when processing it is considered easier/cheaper than processing the actual object. Later the actual object can be reconstituted from the UR. URs are used to represent at LEAST the following in ECS:

- o Data Granules
- o Advertisements
- o Sessions

Application programs that interact with the user will provide URs to users in lieu of the objects the application processes. These URs can be stored in files, e-mailed, etc.

Detailed design documentation can be found in the Release B 305 Overview documentation. Additional material is available from the current maintainer.

These classes are fully thread-safe. Accesses to static data are protected by mutexes.

Classes

- EcUrServerUR -- The EcUrServerUR class, which is inherited from the EcUrUR class, holds additional attributes and overrides EcUrUR methods in order to support Server URs.
- EcUrServerURProvider -- The EcUrServerURProvider class, which is inherited from the EcUrURProvider class, holds additional attributes and overrides EcUrURProvider methods in order to support Server UR providers.
- EcUrUR -- The Universal Reference (UR) is the abstract base class for all references to UR Providers.
- EcUrURMaker -- This class has the responsibility to aid working with objects that subclass UR that are either abstract or whose type is not known.
- EcUrURProvider -- This class is the abstract base class for all things referred to by Universal Reference (UR)s.
- EcUrURProviderMaker -- This class supports the creation of fully reconstituted UR providers given a UR.

library EcUrSh (shared)

/ecs/formal/COMMON/CSCI_UR/src/UR/framework

A Universal Reference (UR) is created as a logical reference to a extant ECS object. This reference can be passed around, written to disk, read from disk, etc. A UR is used when processing it is considered easier/cheaper than processing the actual object. Later the actual object can be reconstituted from the UR. URs are used to represent at LEAST the following in ECS:

- o Data Granules
- o Advertisements
- o Sessions

Application programs that interact with the user will provide URs to users in lieu of the objects the application processes. These URs can be stored in files, e-mailed, etc.

Detailed design documentation can be found in the Release B 305 Overview documentation. Additional material is available from the current maintainer.

These classes are fully thread-safe. Accesses to static data are protected by mutexes.

Classes

- EcUrServerUR -- The EcUrServerUR class, which is inherited from the EcUrUR class, holds additional attributes and overrides EcUrUR methods in order to support Server URs.
- EcUrServerURProvider -- The EcUrServerURProvider class, which is inherited from the EcUrURProvider class, holds additional attributes and overrides EcUrURProvider methods in order to support Server UR providers.
- EcUrUR -- The Universal Reference (UR) is the abstract base class for all references to UR Providers.
- EcUrURMaker -- This class has the responsibility to aid working with objects that subclass UR that are either abstract or whose type is not known.
- EcUrURProvider -- This class is the abstract base class for all things referred to by Universal Reference (UR)s.
- EcUrURProviderMaker -- This class supports the creation of fully reconstituted UR providers given a UR.

library EcUt

/ecs/formal/COMMON/CSCI_Util/src/Logging

EcUt is a utility library for all the ECS applications. It provides following services:

- o Error Logging Mechanism
- o Passing and returning status code
- o Reusable interface for objects which need to be transferred between processes
- o Provides a unique identifier that is guaranteed to be unique within the ECS system.
- o Uniquely identifies any RPC call between a client and a server process anywhere in the ECS enterprise.

Classes

- EcLgErrorEvent -- This object provides a simple mechanism for converting a light-weight uuid_s_ok error message object (EcLgErrorMsg) into an EcAgEvent.
- EcLgErrorMsg -- This object encapsulates an ECS error event in a more lightweight container than the EcAgEvent.
- EcUtErrorDetails -- EcUtErrorDetails is an accessor class for obtaining detailed information about an error code received from downstream.
- EcUtRpcID -- EcUtRpcID is used to uniquely identify any RPC call between a client and a server process anywhere in the ECS enterprise.
- EcUtStreamable -- The EcUtStreamable class provides a reusable interface for y// objects which need to be transferred between processes.
- EcUtUniqueID

library EcUtFactory

/ecs/formal/COMMON/CSCI_Util/src/Factory

is a template class that allows you to specify the base . Accesses to the factory are protected by mutexes. The EcUtFactory should be used when you dynamically create objects based on the a class identifier.

EcUtFactory is a template class that allows you to specify the base class to use as the root of all objects created. In addition, #define macros are provided that allow derived classes to automatically be "register" with the factory with no additional programming.

The EcUtFactory support operations to Register a new class and Make an object of that class.

For those familiar with RogueWave collectibles, this class offers similar capabilities but is lighter-weight because it does not require defining save-guts/restore-guts. It is more flexible in several ways. It is template based allowing for an arbitrary base class. It allows for strings or integers as the classID. It support dynamic registration in a multi-threaded environment (GuardedFactory only).

Classes

- EcUtClassID -- This object allows an class identifier to be created with a value, this identifier to be assigned, compared, and checks if it is valid.

Templates

- EcUtFactory -- This class supports the registering and execution of functions to create subclasses of <BaseClass>.
- EcUtGuardedFactory -- This class supports the creation of subclasses of <BaseClass> in a threaded environment.

library EcUtFactorySh (shared)

/ecs/formal/COMMON/CSCI_Util/src/Factory

The should be used when you dynamically create objects based on the a class identifier.

is a template class that allows you to specify the base class to use as the root of all objects created. In addition, #define macros are provided that allow derived classes to automatically be "register" with the factory with no additional programming.

The support operations to Register a new class and Make an object of that class.

For those familiar with RogueWave collectibles, this class offers similar capabilities but is lighter-weight because it does not require defining save-guts/restore-guts. It is more flexible in several ways. It is template based allowing for an arbitrary base class. It allows for strings or integers as the classID. It support dynamic registration in a multi-threaded environment (GuardedFactory only).

The class is a fully thread-safe version of . Accesses to the factory are protected by mutexes.

Classes

- EcUtClassID -- This object allows an class identifier to be created with a value, this identifier to be assigned, compared, and checks if it is valid.

Templates

- EcUtFactory -- This class supports the registering and execution of functions to create subclasses of <BaseClass>.
- EcUtGuardedFactory -- This class supports the creation of subclasses of <BaseClass> in a threaded environment.

library EcUtGUI

/ecs/formal/COMMON/CSCI_GUI/src

Classes

- EcRwCollectable -- The EcRwCollectable class is an abstract base class derived from Rogue Wave's collectable class. It does not redefine any of the inherited functionality but adds IsInheritedFrom method.
- EcUtDateEntry
- EcUtDateInterval
- EcUtDb1SelBox
- EcUtIntegerTextField
- EcUtListbox
- EcUtRangeCheck
- EcUtRealTextField
- EcUtSelectableItem -- It contains an identifying string and points to a contained object.
- EcUtSelectableItemContainer -- This class is intended to store a list of EcUtSelectableItems.
- EcUtSelectableItemState
- EcUtSetTabChildSaved
- EcUtSetTextWidgetEditable
- EcUtTextField
- EcUtTimeOnlyEntry
- EcUtTimeOnlyInterval
- TimeEntry
- TimeInterval

library EcUtGUI

/ecs/formal/COMMON/CSCI_GUI/src

Classes

- EcRwCollectable -- The EcRwCollectable class is an abstract base class derived from Rogue Wave's collectable class. It does not redefine any of the inherited functionality but adds IsInheritedFrom method.
- EcUtDateEntry
- EcUtDateInterval
- EcUtDb1SelBox
- EcUtIntegerTextField
- EcUtListbox

- EcUtRangeCheck
- EcUtRealTextField
- EcUtSelectableItem -- It contains an identifying string and points to a contained object.
- EcUtSelectableItemContainer -- This class is intended to store a list of EcUtSelectableItems.
- EcUtSelectableItemState
- EcUtSetTabChildSaved
- EcUtSetTextWidgetEditable
- EcUtTextField
- EcUtTimeOnlyEntry
- EcUtTimeOnlyInterval
- TimeEntry
- TimeInterval

library EcUtHelp

/ecs/formal/COMMON/CSCI_HELP/src/HelpClass

Classes

- EcUtCIHelp -- Comment ReadOnlyBTree, use RWBTree instead "..

library EcUtHelp

/ecs/formal/COMMON/CSCI_HELP/src/HelpClass

Classes

- EcUtCIHelp -- Comment ReadOnlyBTree, use RWBTree instead "..

library EcUtMisc

/ecs/formal/COMMON/CSCI_Util/src/Misc

This library contains miscellaneous utilities.

Classes

- EcUtFileCopy -- Objects of this class copy a single file from the specified source location to a specified destination location.
- EcUtFileManager -- This class supports expanding special characters in filenames.
- EcUtPerfData -- Encapsulates performance logging functions.

library EcUtMiscSh (shared)

/ecs/formal/COMMON/CSCI_Util/src/Misc

The EcUtMisc library contains miscellaneous utilities.

Classes

- EcUtFileCopy -- Objects of this class copy a single file from the specified source location to a specified destination location.
- EcUtFileManager -- This class supports expanding special characters in filenames.
- EcUtPerfData -- Encapsulates performance logging functions.

library EcUtNoPF

/ecs/formal/COMMON/CSCI_Util/src/Logging

EcUtNoPF is a utility library for all the ECS applications that do not use Process Framework. It provides following services:

- o Error Logging Mechanism
- o Passing and returning status code
- o Resuable interface for objects which need to be transferred between processes
- o Provides a unique identifier that is guaranteed to be unique within the ECS system.
- o Uniquely identifies any RPC call between a client and a server process anywhere in the ECS enterprise.

Classes

- EcUtErrorDetails -- EcUtErrorDetails is an accessor class for obtaining detailed information about an error code received from downstream.
- EcUtRpcID -- EcUtRpcID is used to uniquely identify any RPC call between a client and a server process anywhere in the ECS enterprise.
- EcUtStreamable -- The EcUtStreamable class provides an reuseable interface for y// objects which need to be transferred between processes.
- EcUtUniqueID

library EcUtSh (shared)

/ecs/formal/COMMON/CSCI_Util/src/Logging

EcUt is a utility library for all the ECS applications. It provides following services:

- o Error Logging Mechanism
- o Passing and returning status code
- o Resuable interface for objects which need to be transferred between processes
- o Provides a unique identifier that is guaranteed to be unique within the ECS system.
- o Uniquely identifies any RPC call between a client and a server process anywhere in the ECS enterprise.

Classes

- EcLgErrorEvent -- This object provides a simple mechanism for converting a light-weight uuid_s_ok error message object (EcLgErrorMsg) into an EcAgEvent.

- `EcLgErrorMsg` -- This object encapsulates an ECS error event in a more lightweight container than the `EcAgEvent`.
- `EcUtErrorDetails` -- `EcUtErrorDetails` is an accessor class for obtaining detailed information about an error code received from downstream.
- `EcUtRpcID` -- `EcUtRpcID` is used to uniquely identify any RPC call between a client and a server process anywhere in the ECS enterprise.
- `EcUtStreamable` -- The `EcUtStreamable` class provides a reusable interface for `Y//` objects which need to be transferred between processes.
- `EcUtUniqueID`

library event

`/ecs/formal/CSS/DOF/src/LOGGING/clientlog`

Purpose of this library is to provide a framework for MSS Agent to store information to the MSS log file and rename the file when the file is full.

Classes

- `EcUtClientLogger` -- `EcUtClientLogger` is used for Application logs.
- `EcUtLoggerRelA` -- This class provides a mechanism for logging information.

library eventSh (shared)

`/ecs/formal/CSS/DOF/src/LOGGING/clientlog`

Purpose of this library is to provide a framework for MSS Agent to store information to the MSS log file and rename the file when the file is full.

Classes

- `EcUtClientLogger` -- `EcUtClientLogger` is used for Application logs.
- `EcUtLoggerRelA` -- This class provides a mechanism for logging information.

library Gl

`/ecs/formal/DSS/sdsrv/src/gl`

This library is used by many subsystems to provide a general purpose list object for storing various scalar and complex datatypes. The list and its elements can be read from or saved to a stream or file. This facilitates exchange of information (parameters) between processes.

Classes

- `GlBinaryP` -- This class provides a specialization of the `GlParameter` for storing Binary data.
- `GlCircleP` -- This class is a specialized parameter that represents a circle.
- `GlClient` -- This is a specialized parameter that represents a client.
- `GlDateP` -- `GlDateP` // specialization of `GlParameter` for dates.
- `GlDoubleP` -- Specialization of `GlParameter` for doubles.

- `GlGPolygonP`
- `GlLocalMemMgr`
- `GlLog` -- Place-holder for full-fledged log.
- `GlLongP` -- Specialization of `GlParameter` for type long.
- `GlParameter`
- `GlParameterBase`
- `GlParameterList` -- This class provides a way to collect and manipulate `Glparameter` objects.
- `GlParameterNameTable` -- This is intended only to be used as a static member object of the `GlParameterBase` class.
- `GlPfxstream`
- `GlPfxstreambuf` -- Specialized streambuf that adds the start-of-line state and emits the given prefix at the beginning of each line.
- `GlPoint` -- Light weight data structure for spatial 2D point.
- `GlPointP` -- This is a specialized parameter that represents a point.
- `GlPointSet` -- This class provides a light weight data structure for a set of 2D points.
- `GlPolygon` -- Light weight data structure for a polygon derived from `GlPointSet`
- `GlPolygonP` -- This class is a specialized parameter that represents a polygon.
- `GlPolygonSet` -- This class is a light weight data structure for a set of polygons.
- `GlPolygonSetP` -- `PolygonSet` as `GlParameter`.
- `GlRectangleP` -- This class is a specialized parameter for rectangle.
- `GlSpatialP` -- This is an abstract base class for spatial objects.
- `GlStringP` -- Specialization of `GlParameter` for strings.
- `GlTimeP` -- Specialization of `GlParameter` for time data.
- `GlTransfer` -- This class contains the specialized `Flatten` and `Restore` operations for `RWCcollectibles`, which are necessary to portably send them across the network to other machines with potentially different architectures.

library GlSh (shared)

/ecs/formal/DSS/sdsrv/src/gl

This library is used by many subsystems to provide a general purpose list object for storing various scalar and complex datatypes. The list and its elements can be read from or saved to a stream or file. This facilitates exchange of information (parameters) between processes.

Classes

- `GlBinaryP` -- This class provides a specialization of the `GlParameter` for storing Binary data.
- `GlCircleP` -- This class is a specialized parameter that represents a circle.
- `GlClient` -- This is a specialized parameter that represents a client.
- `GlDateP` -- `GlDateP` // specialization of `GlParameter` for dates.

- GIDoubleP -- Specialization of GIParameter for doubles.
- GIGPolygonP
- GILocalMemMgr
- GILog -- Place-holder for full-fledged log.
- GILongP -- Specialization of GIParameter for type long.
- GIParameter
- GIParameterBase
- GIParameterList -- This class provides a way to collect and manipulate GIParameter objects.
- GIParameterNameTable -- This is intended only to be used as a static member object of the GIParameterBase class.
- GIPfxstream
- GIPfxstreambuf -- Specialized streambuf that adds the start-of-line state and emits the given prefix at the beginning of each line.
- GIPoint -- Light weight data structure for spatial 2D point.
- GIPointP -- This is a specialized parameter that represents a point.
- GIPointSet -- This class provides a light weight data structure for a set of 2D points.
- GIPolygon -- Light weight data structure for a polygon derived from GIPointSet
- GIPolygonP -- This class is a specialized parameter that represents a polygon.
- GIPolygonSet -- This class is a light weight data structure for a set of polygons.
- GIPolygonSetP -- PolygonSet as GIParameter.
- GIRectangleP -- This class is a specialized parameter for rectangle.
- GISpatialP -- This is an abstract base class for spatial objects.
- GIStringP -- Specialization of GIParameter for strings.
- GITimeP -- Specialization of GIParameter for time data.
- GITransfer -- This class contains the specialized Flatten and Restore operations for RWCollectables, which are necessary to portably send them across the network to other machines with potentially different architectures.

library hdiff

/ecs/formal/PDPS/DPS/SSIT/src/DIF/HDIFF

This library supports the program for the HDF Gui and command line version of the HDF comparison tool.

library InADPP

/ecs/formal/INGEST/ADPP/src/NMC

This library provides additional preprocessing functionality.

Classes

- InGRIBData -- This class defines the high level routines used in the decoding of GRIB data.
- InPpGRIBHDF -- Define the class for preprocessing GRIB HDF data.
- InPpGRIBMetadata -- Define the class for preprocessing GRIB metadata.
- InPpGRIBScienceData -- Define the class for preprocessing GRIB science data.

library InConfig

/ecs/formal/INGEST/common/src/config

This library provides a class which is used by each Ingest executable to obtain parameters from its configuration file.

Classes

- InConfig -- The InConfig class is used by each Ingest executable to obtain parameters from its configuration file.

library InDBaccUt

/ecs/formal/INGEST/common/src/DBaccess

This library provides classes which allows access to the Ingest database.

Classes

- InCurrentDataTypeMap -- The InCurrentDataTypeMap class provides a mapping of current versionID to valid Ingest data types.
- InDataTypeTemplate -- The InDataTypeTemplate class provides the valid Ingest data types.
- InDBAccess -- The InDBAccess class provides access to the Ingest database.
- InDBAccessPool -- The InDBAccessPool class is used by Ingest to obtain a connection to the Ingest database.
- InEDPAddressMap -- The InEDPAddressMap class provides a mapping of external data Provider with given IP address to valid Ingest externaldataproducer.
- InExternalDataProviderInfo -- The InExternalDataProviderInfo class provides the valid Ingest external data provider information.
- InFileTypeTemplate -- The InFileTypeTemplate class provides the file type information for each data type.
- InFileTypeTemplateError -- The InFileTypeTemplateError class is the the error handler for the InFileTypeTemplate class.
- InGranuleServerInfo -- The InGranuleServerInfo class provides the Ingest Granule Server information.
- InMediaType -- The InMediaType class provides the Ingest media types.
- InNextAvailableID -- The InNextAvailableID class keeps track of and issues the next available number for RequestIDs.

- InRequestFileInfo -- The InRequestFileInfo class provides checkpoint storage of file information associated with a given data granule within a given Ingest request.
- InRequestProcessData -- The InRequestProcessData contains real-time process information regarding Data Types within a given request currently being processed or waiting to be processed by the system.
- InRequestProcessHeader -- The InRequestProcessHeader class contains real-time process information regarding Ingest requests currently being processed or waiting to be processed by the system.
- InRequestSummaryData -- The InRequestSummaryData class provides long-term storage of summary data type statistics associated with a given data granule in a given Ingest request.
- InRequestSummaryHeader -- The InRequestSummaryHeader class provides long-term storage of summary request-level statistics associated with a given Ingest request.
- InSourceMCF -- The InSourceMCF class contains metadata extraction information.
- InSourceMCFError -- The InSourceMCFError class is the error handler for the InSourceMCF class.
- InSystemParameters -- The InSystemParameters class provides the Ingest system-level tunable parameters.
- InTDbDataTypeTemplate -- Structure containing the data in row of the DataTypeTemplate table.
- InTDbDDNFileInfo -- Structure containing the DDN data needed for one row of the RequestFileInfo table.
- InValDataGranuleState -- The InValDataGranuleState class provides the valid data granule states.
- InValNotifyType -- The InValNotifyType class provides the valid notify types.
- InValRequestState -- The InValRequestState class provides the valid request states.

library InFTP

/ecs/formal/INGEST/common/src/ftp

This library provides a class with file transfer capabilities.

Classes

- CsFtFTP -- This Class provides file transfer capabilities.

library InGranResource

/ecs/formal/INGEST/Resource/src/resource

This library allocates/deallocates staging disk resources and calls the IngestFtpServer to transfer files.

Classes

- InFile -- It transfers an ingested file.

- InGranResourceIF -- Serves as an interface from Granule Server to resource/device services (e.g., device allocation, data transferring, and file information) which is provided by the Storage Management Subsystem.

library InGranuleC

/ecs/formal/INGEST/GranServer/src/Server

This library supports the client side of the SRF request.

Classes

- InGranuleAsync_C -- The InGranuleAsync_C object provides bi-directional communications with the InGranuleAsync_S object which resides in the server address space.
- InGranuleList -- This class contains granule information pertaining to a DAN request.

library InGranuleMsg

/ecs/formal/INGEST/GranServer/src/Server

This library supports the messages needed for SRF to communicate between Granule Server and Request Manager.

Classes

- InGranuleCompleteMsg
- InGranuleRequestMsg

library InHostInfo

/ecs/formal/INGEST/common/src/hostinfo

This library retrieves host information.

Classes

- EcUtHostInfo -- This class encapsulate few funtion call in the form the socket library.

library InMetadataToolAPI

/ecs/formal/INGEST/ADPP/src/PGS

This library provides the class which accesses the PGS Toolkit from a non PGE C++ environment.

Classes

- InMetadataToolAPI -- This class encapsulates the TOOLKITS "c" language library call and extends the TOOLKIT functionality.

library InMsg

/ecs/formal/INGEST/common/src/Messages

This library provides classes that are used for all data messages exchanged between the external source and Ingest.

Classes

- CsGwTransferPkt_1_0
- InDAN -- The InDAN class contains services to access PVL statements in the DAN message.
- InDDA -- The InDDA class provides services to unpack the DDA message.
- InDDN -- The InDDN class provides services to package the DDN message.
- InDDNTransferPkt_1_0
- InLongDAA -- The InLongDAA class provides services to package the long DAA message.
- InLongDDN -- The InLongDDN class provides services to package the long DDN message.
- InMessage -- The InMessage class defines all data messages exchanged between the external source and Ingest.
- InShortDAA -- The InShortDAA class provides services to package the short DAA message.
- InShortDDN -- The InShortDDN class provides services to package the short DDN message.
- PktStructDef_1_0
- SdpfTransferPkt_1_0

library InPreprocess

/ecs/formal/INGEST/Preprocess/src/code2

This library provides classes to preprocess data by extracting the appropriate metadata for each data type.

Classes

- InBOMetadata -- This class provides the ability to preprocess byte ordered metadata.
- InBUFRMetadata -- This class provides the ability to preprocess parameter value metadata.
- InDataPreprocessList -- The class provides the ability to add files to a list and to get files off the list.
- InDataPreprocessTask -- The Class provides the ability to preprocess ingest files associated with various data types.
- InDataPreprocessTaskError -- Class that handles any error that occurred during preprocessor.
- InDataServerInsertionTask -- The class provides the ability to insert data files into the Document Data Server and Science Data Server.
- InEDOSMetadata -- This class provides the ability to preprocess parameter value metadata.
- InEMOSMetadata -- This class provides the ability to preprocess parameter value metadata.
- InFDDMetadata -- This class provides the ability to preprocess parameter value metadata.

- InFDDorbitMetadata -- This class provides the ability to preprocess byte-ordered FDD Orbit metadata.
- InISCCPMetadata -- This class provides the ability to preprocess byte ordered metadata for the ISCCP data types.
- InL7MOCMetadata -- This class provides the ability to parse the filename that is provided by L7MOC in order to obtain the range beginning ending time/date.
- InMetadata -- The InMetadata class is the abstract base class for other metadata classes such as the InBOMetadata class, InPVMetadata, and InHDFMetadata class.
- InNCEPMetadata -- This class provides the ability to preprocess parameter value metadata.
- InNCEPT62Metadata -- This class provides the ability to preprocess parameter value metadata.
- InODLMetadata -- The class file provides the ability to preprocess parameter ODL metadata.
- InOZONSBMetadata -- This class provides the ability to preprocess byte ordered metadata for the OZON_SB data type.
- InPpCCSDSTime -- This class converts the CCSDS Spacecraft Time Stamp to the Ingest Preprocessing date time string format.
- InPpConvertSOLib -- Define the class to convert SO library.
- InPpDate -- Define the Ingest Preprocessing date class.
- InPpDateTime -- Define the class for Preprocessing date and time.
- InPpDateTimeArray -- Define the class for Preprocessing date and time arrays.
- InPpExternalScienceData -- Define the class for Ingest Preprocessing external science data.
- InPpODLTools -- Define the class for Preprocessing ODL Tools.
- InPpPGSToolkitUtils -- Define the class for the PGS toolkit utilities.
- InPpScienceData -- This class defines the base class InPpScienceData for various science data specializations.
- InPpSharedObjectLibrary -- Define the Ingest Preprocessing shared object library.
- InPpTime -- Define the Ingest Preprocessing time related stuff.
- InPVMetadata -- This class provides the ability to preprocess parameter value metadata.
- InSpecMetadata -- The class file provides the ability to preprocess parameter ODL metadata.
- InVEGIMetadata -- This class provides the ability to preprocess Third Generation Global Vegetation Index metadata.

library InResource

/ecs/formal/INGEST/Resource/src/resource

This library allocates/deallocates staging disk resources.

Classes

- IndataTypeList -- The set of data types associate with the ingest files.

- InResourceIF -- Serves as an interface to resource/device services (e.g., device allocation, data transferring, and file information) which is provided by the Data Server Subsystem.

library InRmQueue

/ecs/formal/INGEST/RequestMgr/src/Request

This library contains code for Request Managers queue.

Classes

- InGranuleServersQueue -- This class defines the singleton queue InGranuleServersQueue used by the Request Manager to keep track of pending requests for all available Granule Servers.
- InPendingQueue -- This class defines the queue InPendingQueue which is a priority queue of pending requests(implemented as InPendingQueueInfo) for a particular Granule Server.
- InPendingQueueInfo -- This class defines the objects which make up the InPendingQueue queue.

library InSnowIce

/ecs/formal/INGEST/ADPP/src/SnowIce

This library provides the class which inserts all of the data sets extracted from the native format file into SDSs within HDF-EOS.

Classes

- InSnowIceData -- The InSnowIceData class provides the operations for extracting and reformatting the data from a NESDIS EDR Masterfile.
- InSnowIceHDFMgr -- The InSnowIceHDFMgr class inserts all of the data sets extracted from the native format file into SDSs with HDF-EOS.

library InSysCmd

/ecs/formal/INGEST/common/src/syscmd

This library provides classes which allows applications to invoke multi-line unix commands.

Classes

- EcUtSysCmd -- Helps execute multiline system commands using sh.

library InTOMS

/ecs/formal/INGEST/ADPP/src/TOMS

This library provides the class that is designed to pre-process TOMS data in CD-ROM format into the standard ECS Internal format.

Classes

- InTOMSData -- This class is used to process TOMS data.

- InTOMSHDF -- This class is used to process TOMS HDF data.

library InUpdateUR

/ecs/formal/INGEST/common/src/ur

This library provides the interface with Advertising to get the the UR associated with each data type.

Classes

- InUpdateUR -- This Class interfaces with Advertising Services to get the UR associated with each Data type.

library IoAdCore

/ecs/formal/IOS/AdvService/src/CoreLib

The CoreLib CSC provides a set of functionalities for describing a generic advertisement, parsing and validating attributes that support the ECS Core meta data model, and requesting and moderating advertisements.

CoreLib provides a set of API for users/contributors to submit requests. There are several types of requests such as addition of new advertisements, modification or deletion of an already existing approved ads. It also provides a set of API for moderators to fetch, store, accept, reject, or save incoming requests. And finally, it provides a set of API for moderation group administration such as fetching, storing, updating, or deleting group. The use of moderator and administrator require proper DCE ACL privileges.

Classes

- IoAdAdvertisement -- This abstract base entity class supports operations to allow the definition of a general advertisement entity.
- IoAdAdvMessage -- This class supports remote execution of operations of objects derived from IoAdAdvertisement objects.
- IoAdApprovedAdv -- This class models the baseline of approved ads.
- IoAdApprovedAdvList -- This class behaves as a list of approved ads with two additional features.
- IoAdApprovedAdvProxy -- This class aids in the implementation of a approved advertisement.
- IoAdApprovedAdvRep -- This class provides DCE-server based persistent operations for an approved advertisement.
- IoAdAttribute -- Store attributes associated with an advertisement.
- IoAdAttributeList -- We are persistent list of attributes with two additional features.
- IoAdAttributeOperations -- Defines member functions for IoAdAttributeOperations class that support submission and searching of new data types.
- IoAdClassID -- This class acts as a convertor between the standard classID and our needs.

- IoAdConfigFile -- This class is used to store all the sections of the config file into the dictionary file.
- IoAdConfigSection -- This class is using to store all the parameters of a section in the config file.
- IoAdDescParser -- Parser utility that reads HTML files in order to generate attribute lists required by Advertisements.
- IoAdFactory -- This class will generate the appropriate objects based on the type.
- IoAdGroup -- Advertisements that are logically related by subject area are all placed in a IoAdGroup, and a given moderator is assigned to moderate all ads in that group.
- IoAdGroupProxy -- This class aids in the implementation of a group of related advertisements.
- IoAdGroupRep -- This class provides DCE-server based persistent operations for a group of advertisements.
- IoAdGroupState -- Embodiment the state and state-related operations for a group of advertisements.
- IoAdParseEvent -- This class models an event that occurs while "parsing".
- IoAdParseEventList -- A list of events that occurred while parsing complex data.
- IoAdRequest -- This class models all requests made of advertising to change the baseline of approved ads.
- IoAdRequestMessage -- We extend the locale processing to handle Accept and Reject operations
- IoAdRequestProxy -- This class aids in the implementation of a request to change the baseline of approved advertisements.
- IoAdRequestRep -- This class provides DCE-server based persistent operations for a request to change the set of approved advertisements.
- IoAdRequestState -- Embodiment the state and state-related operations for a request to add, update or delete an approved ad (IoAdApprovedAdv).

library IoAdCoreSh (shared)

/ecs/formal/IOS/AdvService/src/CoreLib

The CoreLib CSC provides a set of functionalities for describing a generic advertisement, parsing and validating attributes that support the ECS Core meta data model, and requesting and moderating advertisements.

CoreLib provides a set of API for users/contributors to submit requests. There are several types of requests such as addition of new advertisements, modification or deletion of an already existing approved ads. It also provides a set of API for moderators to fetch, store, accept, reject, or save incoming requests. And finally, it provides a set of API for moderation group administration such as fetching, storing, updating, or deleting group. The use of moderator and administrator require proper DCE ACL privileges.

Classes

- IoAdAdvertisement -- This abstract base entity class supports operations to allow the definition of a general advertisement entity.
- IoAdAdvMessage -- This class supports remote execution of operations of objects derived from IoAdAdvertisement objects.
- IoAdApprovedAdv -- This class models the baseline of approved ads.
- IoAdApprovedAdvList -- This class behaves as a list of approved ads with two additional features.
- IoAdApprovedAdvProxy -- This class aids in the implementation of a approved advertisement.
- IoAdApprovedAdvRep -- This class provides DCE-server based persistent operations for an approved advertisement.
- IoAdAttribute -- Store attributes associated with an advertisement.
- IoAdAttributeList -- We are persistent list of attributes with two additional features.
- IoAdAttributeOperations -- Defines member functions for IoAdAttributeOperations class that support submission and searching of new data types.
- IoAdClassID -- This class acts as a convertor between the standard classID and our needs.
- IoAdConfigFile -- This class is used to store all the sections of the config file into the dictionary file.
- IoAdConfigSection -- This class is using to store all the parameters of a section in the config file.
- IoAdDescParser -- Parser utility that reads HTML files in order to generate attribute lists required by Advertisements.
- IoAdFactory -- This class will generate the appropriate objects based on the type.
- IoAdGroup -- Advertisements that are logically related by subject area are all placed in a IoAdGroup, and a given moderator is assigned to moderate all ads in that group.
- IoAdGroupProxy -- This class aids in the implementation of a group of related advertisements.
- IoAdGroupRep -- This class provides DCE-server based persistent operations for a group of advertisements.
- IoAdGroupState -- Embody the state and state-related operations for a group of advertisements.
- IoAdParseEvent -- This class models an event that occurs while "parsing".
- IoAdParseEventList -- A list of events that occurred while parsing complex data.
- IoAdRequest -- This class models all requests made of advertising to change the baseline of approved ads.
- IoAdRequestMessage -- We extend the locale processing to handle Accept and Reject operations

- IoAdRequestProxy -- This class aids in the implementation of a request to change the baseline of approved advertisements.
- IoAdRequestRep -- This class provides DCE-server based persistent operations for a request to change the set of approved advertisements.
- IoAdRequestState -- Embodiment of the state and state-related operations for a request to add, update or delete an approved ad (IoAdApprovedAdv).

library IoAdHtmlCoreSh (shared)

/ecs/formal/IOS/AdvService/src/HtmlCoreLib

The HtmlCoreLib CSC provides a set of interface classes which represent advertisement web pages: advertisement pages and moderation pages.

Classes

- IoAdEsodamPage -- User interface to the IoAdEsodamPageRep class.
- IoAdEsodamPageRep -- This class creates an esod page.
- IoAdEsodPage -- User interface to the IoAdEsodPageRep class.
- IoAdEsodPageRep -- This class creates an esod page.
- IoAdGroupSelection -- This class encapsulates HTML user interface views to selecting an existing group.
- IoAdPage -- User interface to the IoAdPageRep class.
- IoAdPageRep -- This class creates an advertisement page; The constructor sets the contact and mail address.
- IoAdSelection -- This class encapsulates HTML user interface views to selecting an existing advertisement.

library IoAdHtmlSubsSh (shared)

/ecs/formal/IOS/AdvService/src/HtmlSubsLib

The HtmlSubsLib CSC provides a set of interface classes with operations for dynamically constructing HTML pages which represent different types of advertisements: products, services, providers, and approved ads.

Classes

- IoAdAdvertisementView -- This abstract class encapsulates providing a user interface for the IoAdAdvertisement class.
- IoAdApprovedAdvView -- HTML user interface views for creating/updating/displaying Approved Adv.
- IoAdGroupView -- This class encapsulates providing a user interface for the IoAdGroup class.
- IoAdInstallableServiceAdvView -- This class encapsulates providing a user interface for the IoAdInstallable class.

- IoAdMimeTypeServiceView -- This class encapsulates providing a user interface for the IoAdMimeTypeServiceAdv class.
- IoAdProductView -- This class encapsulates providing a user interface for the IoAdProduct class.
- IoAdProviderView -- This class encapsulates providing a user interface for the IoAdProvider class.
- IoAdRequestView -- This class encapsulates providing a user interface for the IoAdRequest class.
- IoAdServerInstall -- This class supports handling all Web Server side tasks for installing MIME and Installable services.
- IoAdServiceView -- This class encapsulates providing a user interface for the IoAdService class.
- IoAdSignatureServiceView -- This class encapsulates providing a user interface for the IoAdSignature ServiceAdv class.
- IoAdApplicationBuilder -- This class supports operations to build advertisement views.

library IoAdInstallBase

/ecs/formal/IOS/AdvService/src/InstallLib

The installer allows one to install a compressed tar file containing some kind of application (modelled through a "IoAdInstallableServiceAdv") or a URL to get to a WWW service (modelled through a "IoAdMimeTypeServiceAdv") onto a users workstation.

When the above mentioned ads are displayed to the user, they contain a hyperlink to a CGI bin program, "IoAdInstallForm". This program then displays a form to allow the user to customize the install process. When the form is complete, the same IoAdInstallForm program is called to process the form. This program requests the Web browser to setup a workstation process "IoAdInstaller" and the server process sends relevant data to the workstation process. The workstation process then can FTP the tar file if this is an application.

In any case the workstation process then calls the ECS Installer developed in the CLS subsystem to actually do the install.

This library contains classes and programs to support the above activities.

Classes

- IoAdFtpWrapper -- This class may be used for automating the interactive program ftp -- the internet standard for file transfer.
- IoAdInstallState -- Class thats contains the shared state between the installer on the workstation and the installer on the server.

library IoAdInstallBaseSh (shared)

/ecs/formal/IOS/AdvService/src/InstallLib

The installer allows one to install a compressed tar file containing some kind of application (modelled through a "IoAdInstallableServiceAdv") or a URL to get to a WWW service (modelled through a "IoAdMimeServiceAdv") onto a users workstation.

When the above mentioned ads are displayed to the user, they contain a hyperlink to a CGI bin program, "IoAdInstallForm". This program then displays a form to allow the user to customize the install process. When the form is complete, the same IoAdInstallForm program is called to process the form. This program requests the Web browser to setup a workstation process "IoAdInstaller" and the server process sends relevant data to the workstation process. The workstation process then can FTP the tar file if this is an application.

In any case the workstation process then calls the ECS Installer developed in the CLS subsystem to actually do the install.

This library contains classes and programs to support the above activities.

Classes

- IoAdFtpWrapper -- This class may be used for automating the interactive program ftp -- the internet standard for file transfer.
- IoAdInstallState -- Class that contains the shared state between the installer on the workstation and the installer on the server.

library IoAdProxy

/ecs/formal/IOS/AdvService/src/ServerLib

The ServerLib CSC provides classes for process framework DCE server and client. In additions, it contains classes for making RPC call to the server from the client. This is the client side library.

Classes

- IoAdConfiguration -- This class provides access to configuration parameters for this subsystem.
- IoAdIdentifiedProxy -- Client processes of the IoAd server contain objects that send messages over the network to perform thier persistence operations.
- IoAdIdentifiedServer -- This class enables proxy/server persistence processing for classes with IDs.
- IoAdLiteClient -- This class supports operations that encapsulates becoming a Client-ProcessFramework-Process.
- IoAdManagedServer -- Class extends the standard managed process-framework server to include requirements for the IoAdServer.
- IoAdMessage -- This is the abstract base class for self addressed synchronous client messages that execute themselves on the server.
- IoAdPersistentMessage -- This class supports remote execution of any of the persistent operations of objects derived from EcPoPersistent.
- IoAdRpc_1_0 -- Interface with a set of remotely-callable procedures.

- IoAdRpc_1_0_ABS -- Interface with a set of remotely-callable procedures.
- IoAdRpc_1_0_Mgr -- Interface with a set of remotely-callable procedures.
- IoAdSession -- This class provides access to specific global objects used by the advertising Core library.

library IoAdProxySh (shared)

/ecs/formal/IOS/AdvService/src/ServerLib

The ServerLib CSC provides classes for process framework DCE server and client. In additions, it contains classes for making RPC call to the server from the client. This is the client side library.

Classes

- IoAdConfiguration -- This class provides access to configuration parameters for this subsystem.
- IoAdIdentifiedProxy -- Client processes of the IoAd server contain objects that send messages over the network to perform thier persistence operations.
- IoAdIdentifiedServer -- This class enables proxy/server persistence processing for classes with IDs.
- IoAdLiteClient -- This class supports operations that encapsulates becoming a Client-ProcessFramework-Process.
- IoAdManagedServer -- Class extends the standard managed process-framework server to include requirements for the IoAdServer.
- IoAdMessage -- This is the abstract base class for self addressed synchronous client messages that execute themselves on the server.
- IoAdPersistentMessage -- This class supports remote execution of any of the persistent operations of objects derived from EcPoPersistent.
- IoAdRpc_1_0 -- Interface with a set of remotely-callable procedures.
- IoAdRpc_1_0_ABS -- Interface with a set of remotely-callable procedures.
- IoAdRpc_1_0_Mgr -- Interface with a set of remotely-callable procedures.
- IoAdSession -- This class provides access to specific global objects used by the advertising Core library.

library IoAdSearch

/ecs/formal/IOS/AdvService/src/SearchLib

The SearchLib CSC provides a set of searching capabilities for approved advertisements, for the administration of groups, and for moderation of requests. It contains a set of API to perform title, text, attributes, date, provider, or group search on approved advertisements.

Classes

- IoAdApprovedAdvSearchCommand -- This class provides interfaces for application to search the set of approved advertisements by specifying criterion.
- IoAdAttrSearchCommand -- This class provides interfaces for application to search the attributes that we using to create the dynamic html page.
- IoAdGroupList -- This class behaves as a list of groups that can stream itself.
- IoAdGroupSearchCommand -- This class provides interfaces for application to search the groups.
- IoAdRequestSearchCommand -- This class provides interfaces for applications to search for requests to change the approved advertisement baseline.
- IoAdSearch -- This class acts as both the client and the server for search related operations.
- IoAdSearchMessage -- This class supports remote execution of operations of objects derived from IoAdSearch.

library IoAdSearchSh (shared)

/ecs/formal/IOS/AdvService/src/SearchLib

The SearchLib CSC provides a set of searching capabilities for approved advertisements, for the administration of groups, and for moderation of requests. It contains a set of API to perform title, text, attributes, date, provider, or group search on approved advertisements.

Classes

- IoAdApprovedAdvSearchCommand -- This class provides interfaces for application to search the set of approved advertisements by specifying criterion.
- IoAdAttrSearchCommand -- This class provides interfaces for application to search the attributes that we using to create the dynamic html page.
- IoAdGroupList -- This class behaves as a list of groups that can stream itself.
- IoAdGroupSearchCommand -- This class provides interfaces for application to search the groups.
- IoAdRequestSearchCommand -- This class provides interfaces for applications to search for requests to change the approved advertisement baseline.
- IoAdSearch -- This class acts as both the client and the server for search related operations.
- IoAdSearchMessage -- This class supports remote execution of operations of objects derived from IoAdSearch.

library IoAdServer

/ecs/formal/IOS/AdvService/src/ServerLib

The ServerLib CSC provides classes for process framework DCE server and client. In additions, it contains classes for making RPC call to the server from the client. This library is the server side library.

Classes

- IoAdConfiguration -- This class provides access to configuration parameters for this subsystem.
- IoAdIdentifiedProxy -- Client processes of the IoAd server contain objects that send messages over the network to perform thier persistence operations.
- IoAdIdentifiedServer -- This class enables proxy/server persistence processing for classes with IDs.
- IoAdLiteClient -- This class supports operations that encapsulates becoming a Client-ProcessFramework-Process.
- IoAdManagedServer -- Class extends the standard managed process-framework server to include requirements for the IoAdServer.
- IoAdMessage -- This is the abstract base class for self addressed synchronous client messages that execute themselves on the server.
- IoAdPersistentMessage -- This class supports remote execution of any of the persistent operations of objects derived from EcPoPersistent.
- IoAdRpc_1_0 -- Interface with a set of remotely-callable procedures.
- IoAdRpc_1_0_ABS -- Interface with a set of remotely-callable procedures.
- IoAdRpc_1_0_Mgr -- Interface with a set of remotely-callable procedures.
- IoAdSession -- This class provides access to specific global objects used by the advertising Core library.

library IoAdServerSh (shared)

/ecs/formal/IOS/AdvService/src/ServerLib

The ServerLib CSC provides classes for process framework DCE server and client. In additions, it contains classes for making RPC call to the server from the client. This library is the server side library.

Classes

- IoAdConfiguration -- This class provides access to configuration parameters for this subsystem.
- IoAdIdentifiedProxy -- Client processes of the IoAd server contain objects that send messages over the network to perform thier persistence operations.
- IoAdIdentifiedServer -- This class enables proxy/server persistence processing for classes with IDs.
- IoAdLiteClient -- This class supports operations that encapsulates becoming a Client-ProcessFramework-Process.
- IoAdManagedServer -- Class extends the standard managed process-framework server to include requirements for the IoAdServer.
- IoAdMessage -- This is the abstract base class for self addressed synchronous client messages that execute themselves on the server.

- IoAdPersistentMessage -- This class supports remote execution of any of the persistent operations of objects derived from EcPoPersistent.
- IoAdRpc_1_0 -- Interface with a set of remotely-callable procedures.
- IoAdRpc_1_0_ABS -- Interface with a set of remotely-callable procedures.
- IoAdRpc_1_0_Mgr -- Interface with a set of remotely-callable procedures.
- IoAdSession -- This class provides access to specific global objects used by the advertising Core library.

library IoAdSubs

/ecs/formal/IOS/AdvService/src/SubsLib

The SubsLib CSC provides a set of derived classes which represent different types of advertisements: products, services, and providers. These classes provide operations to delete, update, insert, and search advertisements.

Classes

- IoAdInstallableServiceAdv -- This entity class supports operations to allow the definition, storage and retrieval of a advertisement of a downloadable service.
- IoAdMimeServiceAdv -- This entity class supports operations to allow the definition, storage and retrieval of a advertisement of a mime service.
- IoAdProduct -- This entity class supports operations to allow the definition, storage and retrieval of a advertisement of a data product.
- IoAdProvider -- This entity class supports operations to allow the definition, storage and retrieval of a advertisement of a data/service provider.
- IoAdService -- This entity class supports operations to allow the definition, storage and retrieval of a advertisement of a service.
- IoAdSignatureServiceAdv -- This entity class supports operations to allow the definition, storage and retrieval of a advertisement of a signature service.

library IoAdSubsSh (shared)

/ecs/formal/IOS/AdvService/src/SubsLib

The SubsLib CSC provides a set of derived classes which represent different types of advertisements: products, services, and providers. These classes provide operations to delete, update, insert, and search advertisements.

Classes

- IoAdInstallableServiceAdv -- This entity class supports operations to allow the definition, storage and retrieval of a advertisement of a downloadable service.
- IoAdMimeServiceAdv -- This entity class supports operations to allow the definition, storage and retrieval of a advertisement of a mime service.
- IoAdProduct -- This entity class supports operations to allow the definition, storage and retrieval of a advertisement of a data product.

- IoAdProvider -- This entity class supports operations to allow the definition, storage and retrieval of a advertisement of a data/service provider.
- IoAdService -- This entity class supports operations to allow the definition, storage and retrieval of a advertisement of a service.
- IoAdSignatureServiceAdv -- This entity class supports operations to allow the definition, storage and retrieval of a advertisement of a signature service.

library MD5Encrypt

/ecs/formal/COMMON/CSCI_Util/src/md5

library MsAcCInt

/ecs/formal/MSS/MCI/Ac/src

The Accountability Management Service makes the user profile available to the various subsystems, such as the Data Server subsystem, information such as the user's electronic mail address and the shipping address, which are used for the distribution of data products ordered. This library provide the APIs to other subsystems.

Classes

- EcAcOrderCMgr -- EcUtStatus Error Code Used .
- EcAcOrderMgr_1_0
- EcAcProfileMgr -- This class represents client side manager for user profile which submit a request to retrieve, insert, update delete a user profile.
- EcAcRegUserMgr -- This class represents client manager for a registered user which submit request to retrieve, create, update, delete a register user.
- EcAcUsrRequestMgr -- This class represents the client side manager to submit a request to MsAcRegUserMgr to retrieve, insert, update, delete a user request.
- MsAcManagedClient
- MsAcRegUserMgr_1_0
- MsAcUsrProfileMgr_1_0
- MsAcUsrRequestMgr_1_0

library MsAcCIntSh (shared)

/ecs/formal/MSS/MCI/Ac/src

The Accountability Management Service makes the user profile available to the various subsystems, such as the Data Server subsystem, information such as the user's electronic mail address and the shipping address, which are used for the distribution of data products ordered. This library provide the APIs to other subsystems.

Classes

- EcAcOrderCMgr -- EcUtStatus Error Code Used .

- EcAcOrderMgr_1_0
- EcAcProfileMgr -- This class represents client side manager for user profile which submit a request to retrieve, insert, update delete a user profile.
- EcAcRegUserMgr -- This class represents client manager for a registered user which submit request to retrieve, create, update, delete a register user.
- EcAcUsrRequestMgr -- This class represents the client side manager to submit a request to MsAcRegUserMgr to retrieve, insert, update, delete a user request.
- MsAcManagedClient
- MsAcRegUserMgr_1_0
- MsAcUsrProfileMgr_1_0
- MsAcUsrRequestMgr_1_0

library MsAcComm

/ecs/formal/MSS/MCI/Ac/src

This library contains class used for both Accountability Server and Client

Classes

- EcAcOrder -- This class represents an EcAcOrder class and get/set method for each attributes.
- EcAcOrderList
- EcAcOrderStatusCode -- This class represents an Order Status Code List retrieve method.
- EcAcOrderStatusCodeList
- MsAcAddress -- This class represents an aggregate of the MsAcUsrProfile and MsAcOrder.
- MsAcError -- This class is used as an exception object.
- MsAcPathFinder -- This function builds the path and filename of accountability related files and CDS names.
- MsAcRegUser
- MsAcRegUserList
- MsAcSybaseTime -- This class get time and date string that can insert into sybase date format.
- MsAcUsrName -- This class represents an aggregate of the MsAcUsrProfile and MsAcOrder.
- MsAcUsrProfile -- This class represents the user profile information.
- MsAcUsrProfileList
- MsAcUsrRequest -- This class represents the user request information.
- MsAcUsrRequestList
- MsAcUsrResUsage -- This class stores system resource usage information for a registered user.

library MsAcCommSh (shared)

/ecs/formal/MSS/MCI/Ac/src

This library contains class used for both Accountability Server and Client

Classes

- EcAcOrder -- This class represents an EcAcOrder class and get/set method for each attributes.
- EcAcOrderList
- EcAcOrderStatusCode -- This class represents an Order Status Code List retrieve method.
- EcAcOrderStatusCodeList
- MsAcAddress -- This class represents an aggregate of the MsAcUsrProfile and MsAcOrder.
- MsAcError -- This class is used as an exception object.
- MsAcPathFinder -- This function builds the path and filename of accountability related files and CDS names.
- MsAcRegUser
- MsAcRegUserList
- MsAcSybaseTime -- This class get time and date string that can insert into sybase date format.
- MsAcUsrName -- This class represents an aggregate of the MsAcUsrProfile and MsAcOrder.
- MsAcUsrProfile -- This class represents the user profile information.
- MsAcUsrProfileList
- MsAcUsrRequest -- This class represents the user request information.
- MsAcUsrRequestList
- MsAcUsrResUsage -- This class stores system resource usage information for a registered user.

library MsAgDeputyGate

/ecs/formal/MSS/MACI/src

This library provides interface to subagent's deputy gate object. This is used by the mode manager GUI and HPOV custom code.

Classes

- EcAgManager -- The EcAgManager is an OODCE distributed object.
- EcAgManagerOD_1_0
- EcAgManagerOD_1_0_ABS
- EcAgManagerOD_1_0_Mgr
- MsAgDeputyGate -- This class is used to by the management server to control most external aspects of the agent.

- MsAgDeputyGateOD_1_0
- MsAgDeputyGateOD_1_0_ABS
- MsAgDeputyGateOD_1_0_Mgr
- MsAgDiscoverer -- This class searches the known configuration directories and updates the static application and program tables base on the products that were installed.
- MsAgEventBufferCollectable -- This class represents a buffer of events.
- MsAgEventHandler -- This class handles event for EcAgManager.
- MsAgEventMgr -- The Event Manager logs events from the monitor.
- MsAgEventMgrOD_1_0
- MsAgEventMgrOD_1_0_ABS
- MsAgEventMgrOD_1_0_Mgr
- MsAgMetricHandler -- This class handles all the reading and creating of metrics for EcAgManager.
- MsAgMgmtBindingHandle -- This class is a base class of the type of object that can be stored in the MsAgMgmtHandleVector.
- MsAgMgmtBindingVector -- This class is an RWOred type that will contain an array of Binding handles to the EcAgManager Client.
- MsAgMgmtHandle -- This class is a specialized MsAgMgmtBindingHandle that connects with an EcAgManager.
- MsAgMgmtIndx
- MsAgMonitor -- Periodically polls performance and status.
- MsAgPerfEvent -- This class represents an event containing performance data.
- MsAgPortMonitor
- MsAgRegistry -- This class is used by the subagent to discover applications and programs installed on the host.
- MsAgRegistryOD_1_0
- MsAgRegistryOD_1_0_ABS
- MsAgRegistryOD_1_0_Mgr
- MsAgSchedTimeMetric
- MsAgScheduleEntry -- It represents one entry in the schedule that contains the function pointer, the time the function is supposed to be executed, arguments for the function in question, and the index of the entry.
- MsAgScheduler -- The scheduler provides the facility to execute a function after constant specified time intervals.
- MsAgSNMPTbl -- The MsAgSNMPTbl represents a physical SNMP table in a MIB.
- MsAgStaticBuffer -- The StaticBuffer is used to store information from the MsAgTblEntry.
- MsAgSubAgentConfig

- MsAgTblEntry -- This is an abstract class representing an entry in an SNMP table.
- MsAgTblMgr -- This class handles tables.
- MsAgTeConfig -- This class is a table entry that stores information about configuration data.
- MsAgTeDynamic -- This class is an abstract class for dynamic table entries.
- MsAgTeDyProc -- Table entry that contains information about a process.
- MsAgTeFault -- This class is a table entry that stores information about fault data.
- MsAgTeIndx -- This class represents an abstract class which contains a metric index, the parent ID, process ID, the level (i.e. app, prog, proc) and a pointer to a handle (i.e. MsAgMgmtHandle).
- MsAgTeMdMg -- This class is an abstract class for mode managed table entries.
- MsAgTePerf -- This class is a table entry that stores information about performance data.
- MsAgTeStApp -- This class embodies the manageable attributes of an ECS application.
- MsAgTeStProg -- This class embodies all the manageable attributes of an ECS program.
- MsAgTeVerExec -- This class is a table entry that stores information about version execution data.

***library* Persistent**

/ecs/formal/DM/GatewayCSCI/src/Persistent

The Persistent library provides the EcDmV0ToEcsGateway with an interface to the data dictionary.

It is used for both the directory and inventory search for resolving queries at a metadata level. The library provides a tailored, abstracted query interface to a database. This is implemented using the DBTools interface of RogueWave.

Classes

- DmAsGwAsterDirResult -- A container class that holds the results of a directory search.
- DmGwAttributeMap -- Its public view consists of two constructors, one taking a collection id for the DDICT and the other taking a ShortName. In each case the other necessary primary key information (InfoMgrName) is retrieved using the configuration information for the .
- DmGwDirectoryQuery -- This class builds a SQL query using the input search parameters and submits it to the Data Dictionary database.
- DmGwDirectoryResult -- A container class that holds the results of a directory search.
- DmGwKeywordSearch -- This class contains the mechanism for extracting keywordNames from the Data Dictionary database for a given set of collectionId and siteId.
- DmGwQuery -- This is an interface class to the Sybase DB library.

library plac

/ecs/formal/PDPS/PLS/PLANG/src/PWB/ac/src

Activities

This library contains the classes of schedulable activities which include basic activity along with its derived objects, such as Disk Activity, Dpr, Pr and Ground Event activities.

Classes

- **PlAcActivity** -- An instance of a PlAcActivity represents a request to change the configuration of a resource.
- **PlAcDiskAct** -- Instances of PlAcDiskAct are activities that are directed specifically at the disk resource.
- **PlAcDpr** -- Instances of PlAcDpr are activities which represent Data Processing Requests.
- **PlAcForceLink** -- Instances of PlAcForceLink are dummy classes, used to trick the linker into linking classes which are usually not known at compile time.
- **PlAcGroundEvent** -- Instances of PlAcGroundEvent are activities which model ground events on resources.
- **PlAcPr** -- Instances of PlAcPr are complex activities which represent Production Requests.

library plcl

/ecs/formal/PDPS/PLS/PLANG/src/PWB/cl/src

Client resource model (rm) messages, and noid.

This library contains inherited classes responsible for setting up client communication with the resource model through specialized messages.

Classes

- **PlClAppl** -- A PlClAppl is an ECS resource model client application class. It is responsible for setting up communication to a resource model and initializing the entire process.
- **PlClForceLink** -- Instances of PlClForceLink are used to force linkage of ECS resource model message classes which may not be present at link time.
- **PlClMsgAlloc** -- PlClMsgAlloc is a request to allocate a specific resource to a specific activity on a specific plan over a specific interval.
- **PlClMsgAllocAct** -- PlClMsgAllocAct is a request to allocate a specific resource to a specific activity on a specific plan over a specific interval.
- **PlClMsgAllocCheck** -- PlClMsgAllocCheck is a SMsgAllocCheck request to check a specific resource to see if a specific activity on a specific plan over a specific interval can be allocated on that resource.
- **PlClMsgAllocMove** -- PlClMsgAllocMove is a request to allocate a specific resource to a specific activity on a specific plan over a specific interval.
- **PlClMsgAllocRqst** -- PlClMsgAllocRqst is a request to allocate a specific resource to a specific activity on a specific plan over a specific interval.

- `PICIMsgCol` -- Instances of `PICIMsgCol` are a message that carries collections from one agent to another.
- `PICIMsgGetActPool` -- Instances of `PICIMsgGetActPool` are messages requesting activities from the resource model.
- `PICIMsgGetActPoolRsp` -- Comprises a response to a request for activities, and carries a copy of all or part of the resource models activity pool.
- `PICIMsgGetPlans` -- Instances of `PICIMsgGetPlans` are a request to get a subset of plans from the resource model.
- `PICIMsgGetPlansRsp` -- Instances of `PICIMsgGetPlansRsp` are a response to a request sent by an `PICIMsgGetPlans`.
- `PICIMsgGetRsNms` -- `PICIMsgGetRsNms` is a request to get the names of all resources known by the resource model.
- `PICIMsgGetRsNmsRsp` -- `PICIMsgGetRsNmsRsp` contains the names of all resources known by the resource model.
- `PICIMsgGroundEvent` -- Instances of `PICIMsgGroundEvent` are messages that have an associated ground event.
- `PICIMsgPlan` -- Instances of `PICIMsgPlan` are messages that have an associated plan.
- `PICIMsgPlanDel` -- Instances of `PICIMsgPlanDel` are messages that tell the resource model to delete the carried plan from its pool.
- `PICIMsgPlanNew` -- Instances of `PICIMsgPlanNew` are a message that tells the resource model about a new plan that has been created.
- `PICIMsgPlanSave` -- Instances of `PICIMsgPlanSave` are messages which tell the resource model to save the currently carried plan in its resource pool.
- `PICIMsgPlanSaveAs` -- Instances of `PICIMsgPlanSaveAs` are messages which tell the resource model save the carried plan under another given name.
- `PICIMsgPlanUpdate` -- Instances of `PICIMsgPlanUpdate` are a message that tells the resource model that a plan has been updated.
- `PICIMsgVCol` -- Instances of `PICIMsgVCol` are a message that carries void collections from one agent to another.
- `PICISrmNoid` -- Instances of `PICISrmNoid` are `HNoids` that manage the finding, requests to, and receiving information from the system `srm` within the era process.

library PICore1

`/ecs/formal/PDPS/PLS/PLANG/src/CoreLib`

This library contains the first half of Planning's core functionality. The `CoreLib` provides functionality for all areas of Planning. It was divided in half due to its large size.

Classes

- `PIConfig` -- This file defines a helper class to manage getting and setting parameters from the process framework configuration file.

- `PIDataGranInsOrdCollectable` -- The sole purpose of this class is to provide a way for other classes in this library to use the functionality of the `RWGranuleCollectable` classes without being derived from `RWCollectable` themselves.
- `PIDataGranule` -- This class describes an individual instance of an ESDT or data file within the PDPS, it may describe either predicted Data Granules, or Granules that are known to exist within the Science Data Server so that they may be.
- `PIDataGranuleCollection` -- This class provides a collection of either static or dynamic granules based on selected criteria.
- `PIDataScheduled` -- This class is a specialization of the `PIPge` class and accounts for PGEs that are scheduled on the condition that the optional output of a previous Dpr exist.
- `PIDataSource`
- `PIDataSourceFactory` -- This class provides simple methods to construct the correct type of Data Source object from the persistent data in the database.
- `PIDataType` -- This class describes a Data Type known to the PDPS and is used to describe the input or output data of a PGE.
- `PIDataTypeReq` -- This class captures the attributes that describe the input relationship between a PGE and a Data Type A.
- `PIDbIfManager` -- This file defines a helper class to manage the various database interfaces.
- `PIDpr` -- This class describes a Data Processing Request.
- `PIDprData` -- This class associates Data-Granules with DPRs.
- `PIOrder` -- This class provides important information about an Ondemand Order.
- `PIOutputYield` -- This class captures the attributes that describe the output relationship between a PGE and a Data Type The class is used when predicting the output data for the PGE that will be produced for a given Data Processing Request.
- `PIPerformance` -- This class describes the performance statistics of PGE.
- `PIPge` -- This class is the base class within a generalization heirachy that describes Product Generation Executives (PGEs) to the PDPS software, it defines the abstract operations required in order that Planning Subsystem can work out when a PGE requires to be
- `PIPgeFactory` -- This class provides simple methods to construct the correct type of PGE object from the persistent data in the database.
- `PIProductionRequest` -- A Production Request describes the need for the scheduling of some PGE to produce a series of Data Granules.
- `PISnapshotScheduled` -- This class is a specialization of the `PIPge` class and accounts for PGEs that are scheduled to process data for regular time period, this class accounts for all the PGEs that can be planned and scheduled in Release A.
- `PITileScheduled` -- This class is a specialization of the `PIPge` class and accounts for PGEs that are scheduled to process data for regular time period for a series of tiles.
- `PITileSchema` -- The `PITileSchema` class is a table used by `PITileScheduled` for getting the spatial definitions of tiles.

- **PITimer** -- This class initiates a timer to alert the subscription manager of the expiration of a wait time for alternat input activation.
- **PITimeScheduled** -- This class is a specialization of the **PIPge** class and accounts for PGEs that are scheduled to process data for regular time period, this class accounts for all the PGEs that can be planned and scheduled in Release A.

library PICore1IF

/ecs/formal/PDPS/PLS/PLANG/src/CoreLib

This library contains the Science Data Server, advertisement, and subscription server interfaces that are needed to run the Production request editor.

Classes

- **PIConfig** -- This file defines a helper class to manage getting and setting parameters from the process framework configuration file.
- **PIDataGranInsOrdCollectable** -- The sole pupose of this class is to provide a way for other classes in this library to use the functionality of the **RWGranuleCollectable** classes without being derived from **RWCollectable** themselves.
- **PIDataGranule** -- This class describes an individual instance of an ESDT or data file within the PDPS, it may describe either predicted Data Granules, or Granules that are known to exist within the Science Data Server so that they may be.
- **PIDataGranuleCollection** -- This class provides a collection of either static or dynamic granules based on selected criteria.
- **PIDataScheduled** -- This class is a specialization of the **PIPge** class and accounts for PGEs that are scheduled on the condition that the optional output of a previous Dpr exist.
- **PIDataSource**
- **PIDataSourceFactory** -- This class provides simple methods to construct the correct type of Data Source object from the persistent data in the database.
- **PIDataType** -- This class describes a Data Type known to the PDPS and is used to describe the input or output data of a PGE.
- **PIDataTypeReq** -- This class captures the attributes that describe the input relationship between a PGE and a Data Type A.
- **PIDbIfManager** -- This file defines a helper class to manage the various database interfaces.
- **PIDpr** -- This class describes a Data Processing Request.
- **PIDprData** -- This class associates Data-Granules with DPRs.
- **PIOrder** -- This class provides important information about an Ondemand Order.
- **PIOutputYield** -- This class captures the attributes that describe the output relationship between a PGE and a Data Type The class is used when predicting the output data for the PGE that will be produced for a given Data Processing Request.
- **PIPerformance** -- This class describes the performance statistics of PGE.

- PIPge -- This class is the base class within a generalization heirachy that describes Product Generation Executives (PGEs) to the PDPS software, it defines the abstract operations required in order that Planning Subsystem can work out when a PGE requires to be
- PIPgeFactory -- This class provides simple methods to construct the correct type of PGE object from the persistent data in the database.
- PIProductionRequest -- A Production Request describes the need for the scheduling of some PGE to produce a series of Data Granules.
- PISnapshotScheduled -- This class is a specialization of the PIPge class and accounts for PGEs that are scheduled to process data for regular time period, this class accounts for all the PGEs that can be planned and scheduled in Release A.
- PITileScheduled -- This class is a specialization of the PIPge class and accounts for PGEs that are scheduled to process data for regular time period for a series of tiles.
- PITileSchema -- The PITileSchema class is a table used by PITileScheduled for getting the spatial definitions of tiles.
- PITimer -- This class initiates a timer to alert the subscription manager of the expiration of a wait time for alternat input activation.
- PITimeScheduled -- This class is a specialization of the PIPge class and accounts for PGEs that are scheduled to process data for regular time period, this class accounts for all the PGEs that can be planned and scheduled in Release A.

library PICore2

/ecs/formal/PDPS/PLS/PLANG/src/CoreLib

This library contains the second half of Planning's core functionality. The CoreLib provides functionality for all areas of Planning. It was divided in half due to its large size.

Classes

- PIAldernate -- 2.9 09/20/96.
- PIComputer -- The PIComputer class, which is inherited from PIResource, .
- PIDASDelta -- This class provides the list of the maximum acceptable deltatimes for each data type used in determining a replan.
- PIDASDifferent
- PIDATRecord -- The PIDATRecord class maintains all the information the predicted availability times of data granules and some of the attributes of the data granules used to generate such predictions.
- PIDbIfManager -- This file defines a helper class to manage the various database interfaces.
- PIDEMOrder -- This class describes a DEM on demand order.
- PIDEMPR -- This class describes a DEM on demand production request.
- PIDPRCollectionNB
- PIHighLevelOrder -- PIHighLevelOrder is a subclass of PIOrder which provides the functionality specific to creating a high level order for ASTER on-Demand.

- PIMetaDataChecks -- These friend functions are needed by DB IF template.
- PINonRoutineArrival -- This class provides a framework for handling data which arrives in a non-routine or random fashion.
- PINonStandardOrder -- This class describes a non-standard L1B on demand order.
- PINonStandardPR -- This class describes a non-standard L1B on demand production request.
- PIONdemandPR -- This class derives from the PIProductionRequest class and provides some additional functionality specific to an ONDemandPR.
- PIOrderFactory -- This class is an abstract factory which creates an instance of the proper PIOrder subclass given the available information.
- PIPGECollection -- This class handles collections of PGEs.
- PIPgeDbQuery -- This class provides functionality to handle specific queries to the PDPS database on behalf of PGEs.
- PIPgePriority -- This class is part of the production strategies which relates a particular PGE to a particular priority.
- PIPrActivator -- This interface class provides attributes and fuction prototypes of all operations that allow the User to add a Production Request (PR) to the list of PR's to be scheduled for activation, schedule DPR's in a given Production Request for activation, and
- PIPrAlternateInputValues -- This class defines an alternate input to a PGE.
- PIPRFactory -- This class is an abstract factory which creates an instance of the proper Production Request subclass given the available information.
- PIPrMetaDataValues -- These friend functions are needed by DB IF template.
- PIProdStrat -- This class describes the production strategies used by the DAAC to determine data processing request priorities.
- PIProfile -- This is a simple class to create a single instance of the client information class required by DSS when using any of it's client interfaces..
- PIPRPriority -- This class describes a table in the PDPS database that matches the types of production requests (on-demand, standard and reprocessing) and the priority assigned to requests of that type.
- PIReplanCriteria -- 1.5 09/17/96.
- PIRescUseThresh -- This class provides methods for evaluating resource usage of a Production Request vis-a-vis the threshold usage values from the PIRescUseThresh table in the PDPS Database.
- PIResource -- This class is the base class in a generalization heirachy describing the production resource.
- PIResourceChange -- This class is used to compare an old resource plan with the current resource plan.
- PIResourceRequirement -- This class defines resource requirements.
- PIRoutineArrival -- Persistent attributes as defined in the schema.

- **PIStandingOrder** -- The class **PIStandingOrder** is created to Support standing order (attached DPR) Ondemand capability.
- **PIStandingOrderGranules** -- The **PIStandingOrderGranules** class save the standing order granule information in this database table after subscription manager determines it is a standing order granule from the notification it received from the SubServer.
- **PIUserParameters** -- This class allows for manipulation of the run time parameters that may be defined for a PGE.
- **PIUserParametersCollection** -- This class defines methods and operations to allow access to all of the User Parameters associated to a particular PGE.
- **PIUserPriority** -- This class is part of the production strategies which relates a particular user to a particular priority.

library pldb

/ecs/formal/PDPS/PLS/PLANG/src/PWB/db/src

Database Interface.

This library contains classes which build the database interface for ECS specific processes. Its classes are responsible for reading, writing from/to the database and construnting the corresponding objects.

Classes

- **PIDbDatabaseCfg** -- **PIDbDatabaseCfg** serves as a base class for all HCL database interface.
- **PIDbDatabaseInfo** -- **PIDatabaseInfo** serves as an encapsulation of the database schema and stores database login information.
- **PIDbDprCfg** -- **PIDbDprCfg** allows loading and saving data processing request (DPR) activities (**PIAcDpr**) from and to the database tables.
- **PIDbGroundEventCfg** -- **PIDbGroundEventCfg** allows loading and saving ground event activities (**PIAcGroundEvent**) from and to the database tables.
- **PIDbPlanCfg** -- **PIDbPlanCfg** allows loading and saving of Plan information (**PIPIPlan**) from and to the database tables.
- **PIDbPrCfg** -- **PIDbPrCfg** allows loading and saving Production Request activities (**PIAcPr**) from and to the database tables.
- **PIDbResourceCfg** -- **PIDbResourceCfg** allows loading the resource pool from the database tables and saving to the database tables from the resource pool.

library pldi

/ecs/formal/PDPS/PLS/PLANG/src/PWB/di/src

Common (motif) components.

This library contains inherited classes responsible for implementing display capabilities for the client application.

Classes

- `PIDiAppl` -- An instance of this class is a type of resource model client display application.
- `PIDiDateTimePanel` -- Instances of `PIDiDateTimePanel` are displayable panels of MOTIF widgets for the entry or display of date values in a "yy/ddd" format and time values in a "<hours>::<minutes>::<seconds>" format. Seconds can be entered or displayed to 1 millisecond ..
- `PIDiMotifNotifier` -- Instances of `PIDiMotifNotifier` are Motif Notifiers for client display application which handle the watch set behavior.

library PIGUIError

/ecs/formal/PDPS/PLS/PLANG/src/GUIError

This library contains the objects needed for SSIT and QA Monitor GUIs to display error messages. PLS is not using this library.

Classes

- `DpAtMessageDialog` -- This class provides the text messages for EcsError GUI popup message boxes.
- `EcsError`

library plip

/ecs/formal/PDPS/PLS/PLANG/src/PWB/ip/src

Interprocess Communication Agent.

This library contains classes which implement PLS communication among different processes.

Classes

- `PIIpAgent` -- Instances of `PIIpAgent` are `HNoidAgents` that have chosen a particular `NameServAbs` implementation, register their `HAddress` with said `NameServAbs`, and interact with the global `HSigMgr` to install a `HNsSigHand` to take over in the event of disaster.

library plmi

/ecs/formal/PDPS/PLS/PLANG/src/PWB/mi/src

Miscellaneous

This library contains miscellaneous classes that are useful to multiple processes. The class in this library is the base class for all applications.

Classes

- `PLMiAppl` -- Instances of `PLMiAppl` are base application classes for all ECS applications.
- `PLMiTextConvertor` -- Instances of `PLMiTextConvertor` are functionoids responsible for string-to-binary and binary-to-string conversions of integer, floating and other binary values.
- `PLMiTimeCvtr` -- Instances of `PLMiTimeCvtr` provides additional time conversion operations.

library PIODMgrClient

/ecs/formal/PDPS/PLS/PLANG/src/OdMgr

This library contains the objects needed for the EcPIODMgr application. The EcPIODMgr server utility is one of the components of the ASTER OnDemand processing functionality.

Classes

- PIODMsgDObj_1_0
- PIODMsgProxy -- Class.

library plpl

/ecs/formal/PDPS/PLS/PLANG/src/PWB/pl/src

Plans.

This library represents the concepts of plans. Plans are collections of scheduled production activities on available resources.

Classes

- PIPIForceLink -- Instances of PIPIForceLink are used to force linkage of ECS plan classes which may not be present at link time.
- PIPIPlan -- Instances of the class PIPIPlan are the production plans.

library plrc

/ecs/formal/PDPS/PLS/PLANG/src/PWB/rc/src

Resources.

This library contains ECS specific resource related classes. Resources represent the actual physical or virtual entities that perform the production runs. These resources are computers, disks, strings, etc. They maintain a representation of their state through time running state lists.

Classes

- PIRcActIdFact -- Instances of PIRcActIdFact are activity ID generators.
- PIRcActState -- Instances of PIRcActState model the state of a resource during which that resource is in the process of executing an activity.
- PIRcAutosys -- Resources to hold a place in the pool for the autosy(s).
- PIRcComputer -- Instances of PIRcComputer are multi-processors with a known configuration of associated disks.
- PIRcCpu -- Instances of PIRcCpu are processors associated with a particular computer and are of a certain type (Sun, SGI, etc.).
- PIRcDataArrivalState -- Instances of PIRcDataArrivalState models the current data arrival state of a resource.
- PIRcDataServer -- Instances of PIRcDataServer provide data server information.

- **PIRcDisk** -- Instances of **PIRcDisk** are computer resources capable of providing disk storage and disk capacity information.
- **PIRcForceLink** -- Instances of **PIRcForceLink** are used to force linkage of ECS resource model classes which may not be present at link time.
- **PIRcIdFactAbs** -- Instances of **PIRcIdFactAbs** are abstract base class id generators.
- **PIRcLocalDisk** -- Instances of **PIRcLocalDisk** are Disk storage units which are "Locally" or directly attached to a Computer and are therefore owned by that computer.
- **PIRcMultiActState** -- This class is a **RRsState** that can keep track of associations between it and multiple **RActivity** 'tasking' requests.
- **PIRcProcessState** -- Instances of **PIRcProcessState** are classes that represent the current processing state of the CPU.
- **PIRcResource** -- Instances of **PIRcResource** are classes from which all other Planning Workbench specific resource classes inherit.
- **PIRcResourceIdFact** -- Instances of **PIRcResourceIdFact** are activity ID generators.
- **PIRcState** -- Instances of **PIRcState** are generic activity states which simply contains the name of the generic state.
- **PIRcStorageState** -- Instances of **PIRcStorageState** are states for storage utilization for a disk.
- **PIRcString** -- Instances of **PIRcString** are logical groupings of computers.

library plrm

/ecs/formal/PDPS/PLS/PLANG/src/PWB/rm/src

Resource Model.

The resouce model is the heart of the Production Planning tool. It is a representation of database in memory, and the interface for reading and updating to the database, and a maintainer for an up-to-date representation of the states of resource, plans and activities in the system. It is also the hub of message switching process.

Classes

- **PIRmAppl** -- Instances of **PIRmAppl** are resource model applications.
- **PIRmCatalog** -- An instance of **PIRmCatalog** is owned by a **PIRmSrm** client.
- **PIRmSigHand** -- Instances of **PIRmSigHand** are signal catchers for a resource model to detect termination signals and save our resource model states.
- **PIRmSrm** -- Instances of **PIRmSrm** are resource model editors that know about ecs notions, plan accesses, etc.
- **PIRmSrmNoid** -- The **PIRmSrmNoid** is an ipc msgAgent that services resource model requests.

library PIRpAc

/ecs/formal/PDPS/PLS/PLANG/src/RPL/ac

Manages the scheduling activities.

Classes

- **PIRpAcActivity** -- An instance of a **PIRpAcActivity** represents a request to change the configuration of a resource.
- **PIRpAcAlloc** -- An instance of a **PIRpAcAlloc** is an allocation containing information as to when the allocation was created, and any parameters that have been modified from the default parameters.
- **PIRpAcDpr** -- Instances of **PIRpAcDpr** are activities which represent Data Processing Requests.
- **PIRpAcForceLink** -- Instances of **PIRpAcForceLink** are dummy classes, used to trick the linker into linking classes which are usually not known at compile time.
- **PIRpAcGroundEvent** -- Instances of **PIRpAcGroundEvent** are activities which model ground events on resources.
- **PIRpAcInterval** -- This class defines attributes and operations [get/set] for **PIRpAcInterval**.
- **PIRpAcResourceReservation** -- This class defines attributes and operations for **PIRpAcResourceReservation**.
- **PIRpAcSchedInfo** -- Instances of **PIRpAcSchedInfo** are objects that contain information related to how an activity will/was scheduled.
- **PIRpAcTimeInfo** -- Instances of **PIRpAcTimeInfo** are base class objects for scheduling time information.

library **PIRpCI**

/ecs/formal/PDPS/PLS/PLANG/src/RPL/cl

Manages the resource model clients.

Classes

- **PIRpCIAppl** -- A **PIRpCIAppl** is an ECS resource model client application class. It is responsible for setting up communication to a resource model and initializing the entire process.
- **PIRpCIForceLink** -- Instances of **PIRpCIForceLink** are used to force linkage of ECS resource model message classes which may not be present at link time.
- **PIRpCIMsgAlloc** -- **PIRpCIMsgAlloc** is a request to allocate a specific resource to a specific activity on a specific plan over a specific interval.
- **PIRpCIMsgAllocAct** -- **PIRpCIMsgAllocAct** is a request to allocate a specific resource to a specific activity on a specific plan over a specific interval.
- **PIRpCIMsgAllocCheck** -- **PIRpCIMsgAllocCheck** is a **SMsgAllocCheck** request to check a specific resource to see if a specific activity on a specific plan over a specific interval can be allocated on that resource.
- **PIRpCIMsgAllocMove** -- **PIRpCIMsgAllocMove** is a request to allocate a specific resource to a specific activity on a specific plan over a specific interval.

- `PIRpCIMsgAllocRqst` -- `PIRpCIMsgAllocRqst` is a request to allocate a specific resource to a specific activity on a specific plan over a specific interval.
- `PIRpCIMsgCol` -- Instances of `PIRpCIMsgCol` are a message that carries collections from one agent to another.
- `PIRpCIMsgGetActPool` -- Instances of `PIRpCIMsgGetActPool` are messages requesting activities from the resource model.
- `PIRpCIMsgGetActPoolRsp` -- Comprises a response to a request for activities, and carries a copy of all or part of the resource models activity pool.
- `PIRpCIMsgGetPlans` -- Instances of `PIRpCIMsgGetPlans` are a request to get a subset of plans from the resource model.
- `PIRpCIMsgGetPlansRsp` -- Instances of `PIRpCIMsgGetPlansRsp` are a response to a request sent by an `PIRpCIMsgGetPlans`.
- `PIRpCIMsgGetRsNms` -- `PIRpCIMsgGetRsNms` is a request to get the names of all resources known by the resource model.
- `PIRpCIMsgGetRsNmsRsp` -- `PIRpCIMsgGetRsNmsRsp` contains the names of all resources known by the resource model.
- `PIRpCIMsgPlan` -- Instances of `PIRpCIMsgPlan` are messages that have an associated plan.
- `PIRpCIMsgPlanNew` -- Instances of `PIRpCIMsgPlanNew` are a message that tells the resource model about a new plan that has been created.
- `PIRpCIMsgPlanSave` -- `PIRpCIMsgPlanSave` is a request to save a plan to a file whose name is specified at the command line.
- `PIRpCIMsgRsChg` -- Instances of `PIRpCIMsgRsChg` are a message that tells the resource model about a modified resource.
- `PIRpCIMsgRsDel` -- Instances of `PIRpCIMsgRsChg` are a message that tells the resource model about a deleted resource.
- `PIRpCIMsgRsNew` -- Instances of `PIRpCIMsgRsNew` are a message that tells the resource model about a new resource.
- `PIRpCIMsgVCol` -- Instances of `PIRpCIMsgVCol` are a message that carries void collections from one agent to another.
- `PIRpClSrmNoid` -- Instances of `PIRpClSrmNoid` are `HNoids` that manage the finding, requests to, and receiving information from the system `srm` within the era process.

library PIRpDb

/ecs/formal/PDPS/PLS/PLANG/src/RPL/db

Accesses and modifies the database.

Classes

- `PIRpDbCommentsCfg` -- `PIRpDbCommentsCfg` class allows loading the comments for a resource from the database tables and saving the comment to the database tables from the GUI.

- `PIRpDbDatabaseCfg` -- `PIRpDbDatabaseCfg` class serves as a base class for all HCL database interface classes.
- `PIRpDbDatabaseInfo` -- `PIDatabaseInfo` class serves as an encapsulation of the database schema and stores database login information.
- `PIRpDbDprCfg` -- `PIRpDbDprCfg` allows loading and saving data processing request (DPR) activities (`PIRpAcDpr`) from and to the database tables.
- `PIRpDbGroundEventCfg` -- `PIRpDbGroundEventCfg` allows loading and saving ground event activities (`PIRpAcGroundEvent`) from and to the database tables.
- `PIRpDbMMSRsCfg` -- The `PIRpDbMMSRsCfg` class contains the control identification values from the configuration file received from Baseline Manager (MSS).
- `PIRpDbPlanCfg` -- `PIRpDbPlanCfg` allows loading and saving of Plan information (`PIPIPlan`) from and to the database tables.
- `PIRpDbReservationCfg` -- `PIRpDbReservationCfg` allows loading and saving reservation activities (`PIRpAcResourceReservation`) from and to the database tables.
- `PIRpDbResourceCfg` -- `PIRpDbResourceCfg` class allows loading the resource pool from the database tables and saving to the database tables from the resource pool.
- `PIRpDbRsCfg` -- The `PIRpDbRsCfg` class is the interface between Resource Planning and Baseline Manager (MSS).

library PIRpDi

/ecs/formal/PDPS/PLS/PLANG/src/RPL/di

A resource model client display application.

Classes

- `PIRpDiAppl` -- An instance of this class is a type of resource model client display application.

library PIRpRc

/ecs/formal/PDPS/PLS/PLANG/src/RPL/rc

Manages resources.

Classes

- `PIRpRcActIdFact` -- Instances of `PIRpRcActIdFact` are activity ID generators.
- `PIRpRcActState` -- Instances of `PIRpRcActState` model the state of a resource during which that resource is in the process of executing an activity.
- `PIRpRcAllocIdFact` -- Instances of `PIRpRcAllocIdFact` are allocation ID generators.
- `PIRpRcAutosys` -- Instances of `PIRpRcAutosys` are logical groupings of strings.
- `PIRpRcComputer` -- Instances of `PIRpRcComputer` are multi-processors with a known configuration of associated disks.
- `PIRpRcCpu` -- Instances of `PIRpRcCpu` are processors associated with a particular computer and are of a certain type (Sun, SGI, etc.).

- **PIRpRcDisk** -- Instances of **PIRpRcDisk** are computer resources capable of providing disk storage and disk capacity information.
- **PIRpRcForceLink** -- Instances of **PIRpRcForceLink** are used to force linkage of ECS resource model classes which may not be present at link time.
- **PIRpRcIdFactAbs** -- Instances of **PIRpRcIdFactAbs** are abstract base class id generators.
- **PIRpRcLocalDisk** -- Instances of **PIRpRcLocalDisk** are Disk storage units which are "Locally" or directly attached to a Computer and are therefore owned by that computer.
- **PIRpRcMultiActState** -- This class is a **RRsState** that can keep track of associations between it and multiple **RActivity** 'tasking' requests.
- **PIRpRcNetworkDisk** -- Instances of **PIRpRcNetworkDisk** are Disk storage units which are "Network Attached" and behave like NFS mounted disks.
- **PIRpRcProcessState** -- Instances of **PIRpRcProcessState** are classes that represent the current processing state of the CPU.
- **PIRpRcRealComputer** -- Instance of **PIRpRcRealComputer** is a physical collection of virtual computers.
- **PIRpRcResource** -- Instances of **PIRpRcResource** are base class for string, computer and disk resources.
- **PIRpRcResourceIdFact** -- Instances of **PIRpRcResourceIdFact** are unique activity ID generators.
- **PIRpRcState** -- Instances of **PIRpRcState** are generic activity states which simply contains the name of the generic state.
- **PIRpRcStorageState** -- Instances of **PIRpRcStorageState** are states for storage utilization for a disk.
- **PIRpRcString** -- Instances of **PIRpRcString** are logical groupings of computers.

library PIRpRm

/ecs/formal/PDPS/PLS/PLANG/src/RPL/srm

Manages the resource model.

Classes

- **PIRpRmAppl** -- Instances of **PIRpRmAppl** are resource model applications.
- **PIRpRmCatalog** -- An instance of **PIRpRmCatalog** is owned by a **PIRpRmSrm** client.
- **PIRpRmSigHand** -- Instances of **PIRpRmSigHand** are signal catchers for a resource model to detect termination signals and save our resource model states.
- **PIRpRmSrm** -- Instances of **PIRpRmSrm** are resource model editors that know about ecs notions, plan accesses, etc.
- **PIRpRmSrmNoid** -- The **PIRpRmSrmNoid** is an ipc msgAgent that services resource model requests.

library PIRpSc

/ecs/formal/PDPS/PLS/PLANG/src/RPL/sc

Manages schedulable resources.

Classes

- PIRpScAllImpct -- Instances of PIRpScAllImpct add the knowledge of oversubscription to the impact scheduler.
- PIRpScComputer -- Instances of PIRpScComputer are schedulable computer resources.
- PIRpScCpu -- Instances of PIRpScCpu are schedulable CPU resources.
- PIRpScForceLink -- Instances of PIRpScForceLink are used to force linkage of schedule resource model classes which may not be present at link time.
- PIRpScLocalDisk -- Instances of PIRpScLocalDisk are schedulable local disk resources.
- PIRpScResource -- This class sets up protocol for all ECS schedulable resources.

library PIRpSi

/ecs/formal/PDPS/PLS/PLANG/src/RPL/si

Manages the Resource Scheduler GUI.

Classes

- PIRpSiActType -- PIRpSiActType is the same of one row in ACTIVITY_TYPE table.
- PIRpSiAppl -- Instance of PIRpSiAppl is derived from PIRpDiAppl.
- PIRpSiCloseWin -- An instance of this class display reservation that has approved status.
- PIRpSiEditWin -- An instance of this class provides a view of the details associated with the selected reservation resource.
- PIRpSiIntervalSelWin -- (Show Reservation's unSelected and Selected intervals windows).
- PIRpSiMsgBox -- An instance of this provides a message popup box for displaying information to the user.
- PIRpSiNoid -- I'm a noid who is only interested in handling resource related SrmClient messages..
- PIRpSiOppGen -- Instances of PIRpSiOppGen are derived from SsiOppGen.
- PIRpSiResourceSelWin -- An instance of this class shows those resources available and associated with the reservation.
- PIRpSiScheduler -- The PIRpSiScheduler is a generic Scheduling System user interface.
- PIRpSiWin -- An instance of the class is a Motif window to setup and display the reservation plan.
- PIRpSiWinAbs -- An instance of this class shares static attributes for all windows.

library PIRpTI

/ecs/formal/PDPS/PLS/PLANG/src/RPL/tl

Manages the scheduled Resources' Timeline GUI.

Classes

- PIRpTlApDefsCfg -- Instances of PIRpTlApDefsCfg handle configuration files for default display resource settings for the timeline.
- PIRpTlAppl -- PIRpTlAppl is the main application class for the timeline.
- PIRpTlColChgList -- Instances of PIRpTlColChgList are lists which are part of the color mapper dialog.
- PIRpTlColChgSt -- Instances of PIRpTlColChgSt map colors to types.
- PIRpTlColChngr -- Instances of PIRpTlColChngr can be used to specify colors for types.
- PIRpTlConfig -- PIRpTlConfig is the base class for all configuration files.
- PIRpTlDefsConfig -- Instances of PIRpTlDefsConfig load and save basic timeline configuration files.
- PIRpTlFileDiag -- An instance of the class is a base dialog for loading and saving the configuration files.
- PIRpTlFileSelDiag -- This is a derived DFileSelectBox that AUTOMATICALLY creates a FileSelectionDialog (i.e. both the FileSelectionBox widget and its DialogShell parent are created no matter what create function you call.) In addition, this class adds behavior to make
- PIRpTlMainRegion -- A PIRpTlMainRegion is a _TLMainRegion_ that implements "timeline" mouse tracking behavior.
- PIRpTlPanel -- This class, which is derived from TLPanel, adds capabilities for interactively changing colors and resources, and for saving and loading from different configuration files.
- PIRpTlPlanWinChgr -- Instances of PIRpTlPlanWinChgr are dialogs for selecting a plan.
- PIRpTlRsConfig -- Instances of PIRpTlRsConfig are responsible for loading a set of names of resources to be displayed on the timeline.
- PIRpTlRsEditor -- Instances of PIRpTlRsEditor are dialogs for choosing the set of resources to be displayed on the timeline.
- PIRpTlSimpleEvent -- Instances of PIRpTlSimpleEvent provide base class behavior that all timeline simple events will provide.
- PIRpTlSliderPup -- Instances of PIRpTlSliderPup are popup windows which display a start time and stop time for the corresponding plan region.
- PIRpTlSrmNoid -- Instances of PIRpTlSrmNoid are timeline resource model client noids.
- PIRpTlStatusBar -- Instances of PIRpTlStatusBar provide a line at the bottom of the timeline for displaying messages.
- PIRpTlStplConfig -- Instances of PIRpTlStplConfig handle stipple keywords in configuration files.
- PIRpTlStplEv -- Instances of PIRpTlStplEv are stipple events.

- `PIRpTlStplEvent` -- Instances of `PIRpTlStplEvent` are stipple events associated with resource states.
- `PIRpTlStplView` -- Instances of `PIRpTlStplView` are views associated with stipple events.
- `PIRpTlStplVw` -- Instances of `PIRpTlStplVw` are views of stipple events.
- `PIRpTlTypesConfig` -- Instances of `PIRpTlTypesConfig` handle configuration files which can contain class names which are added to the list of the timeline's known types..
- `PIRpTlValEvent` -- This class is a derived `TLEvent` which is used for displaying values versus time data graphically on a `TLEventSubRegion` over a specific interval of time.
- `PIRpTlValView` -- This class is the base class for views that display value versus time information graphically in a timeline subregion.
- `PIRpTlWindow` -- `PIRpTlWindow` are main windows used by the `PIRpTlAppl` class.
- `PIRpTlXtl` -- Instances of `PIRpTlXtl` are planning timeline displays.
- `PIRpTlZmBar` -- Instances of `PIRpTlZmBar` are zoom bars which know how to pop up a plan interval window.

library plsc

/ecs/formal/PDPS/PLS/PLANG/src/PWB/sc/src

Schedulable Resources.

The Schedulable resources, in fact, perform the scheduling of the activities on resources. For each resource, there must be a corresponding schedulable resource with it.

Classes

- `PlScAllImpct` -- Instances of `PlScAllImpct` add the knowledge of oversubscription to the impact scheduler.
- `PlScComputer` -- Instances of `PlScComputer` are schedulable computer resources.
- `PlScCpu` -- Instances of `PlScCpu` are schedulable CPU resources.
- `PlScForceLink` -- Instances of `PlScForceLink` are used to force linkage of schedule resource model classes which may not be present at link time.
- `PlScLocalDisk` -- Instances of `PlScLocalDisk` are schedulable local disk resources.
- `PlScResource` -- This class sets up protocol for all ECS schedulable resources.

library PISSIT

/ecs/formal/PDPS/PLS/PLANG/src/CoreLib

This library manages the Science Software Integration & Test PDPS DB interfaces.

Classes

- `PlAlternateCollection` -- It allow for all `PlAlternate` database records associated with a given `PgeId` to be retrieved from the database.

- `PIDataTypeFiles` -- This class captures the attributes that describe the relationship between Names of File Types (within a granule) and the integer value of that file type used by PDPS.
- `PIDataTypeFilesCollection` -- This class constructs a collection of parameters for a Pge given a specific `DataTypeId`.
- `PIDataTypeReqCollection` -- This class constructs a collection of parameters for a given Pge based on a specified `PgeId`.
- `PIDbIfManager` -- This file defines a helper class to manage the various database interfaces.
- `PIExitDependency` -- This class provides access to the name/version (`SswId`), exit code, and condition of another PGE on which this PGE depends.
- `PIExitDependencyCollection` -- This class provides a list of names/versions (`SswIDs`), exit codes, and conditions of PGEs on which this PGE depends.
- `PIExitMessage` -- This class creates descriptive messages which are associated with exit codes generated after PGE processing.
- `PIExitMessageCollection` -- This class provides a list of PGE exit code messages for a given PGE name/version (`SswId`).
- `PIMetaDataChecksCollection` -- This class provides lists of metadata parameter values for PGEs.
- `PIOutputYieldCollection` -- This class selects all the parameters from the database that are associated with a given `PgeId` and constructs a collection of these parameters.

library pltl

/ecs/formal/PDPS/PLS/PLANG/src/PWB/tl/src

Timeline.

This library contains classes responsible for buiding the Timeline application which helps visualize the scheduled activities in a particular plan. Almost all the classes are for building GUI, except the application and noid classes.

Classes

- `PITlApDefsCfg` -- Instances of `PITlApDefsCfg` handle configuration files for default display resource settings for the timeline.
- `PITlAppl` -- An `PITlAppl` is the main application class for the timeline.
- `PITlColChgList` -- Instances of `PITlColChgList` are lists which are part of the color mapper dialog.
- `PITlColChgSt` -- Instances of `PITlColChgSt` map colors to types.
- `PITlColChngr` -- Instances of `PITlColChngr` can be used to specify colors for types.
- `PITlConfig` -- `PITlConfig` is the base class for all configuration files.
- `PITlDefsConfig` -- Instances of `PITlDefsConfig` load and save basic timeline configuration files.
- `PITlFileDiag` -- Instances of `PITlFileDiag` are base classes dialogs for handling the loading and saving of configuration files.

- PITIFileSelDiag -- This is a derived DFileSelectBox that creates a FileSelectionDialog (i.e. both the FileSelectionBox widget and its DialogShell parent are created no matter what create function you call.) In addition, this class adds behavior to make sure the parent
- PITILoadDiag -- Instances of PITILoadDiag are dialogs for loading saved timeline configuration files.
- PITIMainRegion -- A PITIMainRegion is a _TLMainRegion_ that implements "timeline" mouse tracking behavior.
- PITIPanel -- This class, which is derived from TLPanel, adds capabilities for interactively changing colors and resources, and for saving and loading from different configuration files.
- PITIPlanSelDiag -- This is a derived DSelectionBox that creates a PlanSelectionDialog (i.e. both the SelectionBox widget and its DialogShell parent are created no matter what create function you call.) In addition, this class adds behavior to make sure the DialogShell
- PITIPlanWinChgr -- Instances of PITIPlanWinChgr are dialogs for selecting a plan.
- PITIRsConfig -- Instances of PITIRsConfig are responsible for loading a set of names of resources to be displayed on the timeline.
- PITIRsEditor -- Instances of PITIRsEditor are dialogs for choosing the set of resources to be displayed on the timeline.
- PITISaveDiag -- Instances of PITISaveDiag are dialogs for saving timeline configuration files.
- PITISimpleEvent -- Instances of PITISimpleEvent provide base class behavior that all timeline simple events will provide.
- PITISliderPup -- Instances of PITISliderPup are popup windows which display a start time and stop time for the corresponding plan region.
- PITISrmNoid -- Instances of PITISrmNoid are timeline resource model client noids.
- PITIStatusBar -- Instances of PITIStatusBar provide a line at the bottom of the timeline for displaying messages.
- PITIStplConfig -- Instances of PITIStplConfig handle stipple keywords in configuration files.
- PITIStplEv -- Instances of PITIStplEv are stipple events.
- PITIStplEvent -- Instances of PITIStplEvent are stipple events associated with resource states.
- PITIStplView -- Instances of PITIStplView are views associated with stipple events.
- PITIStplVw -- Instances of PITIStplVw are views of stipple events.
- PITITypesConfig -- Instances of PITITypesConfig handle configuration files which can contain class names which are added to the list of the timeline's known types..
- PITIValEvent -- This class is a derived TLEvent which is used for displaying values versus time data graphically on a TLEventSubRegion over a specific interval of time.
- PITIValView -- This class is the base class for views that display value versus time information graphically in a timeline subregion.
- PITIWindow -- PITIWindow are main windows used by the PITIAppl class.
- PITIXtl -- Instances of PITIXtl are planning timeline displays.

- PITlZmBar -- Instances of PITlZmBar are zoom bars which know how to pop up a plan interval window.

library PIUtil

/ecs/formal/PDPS/PLS/PLANG/src/CoreLib

This library contains utilities dealing with current time, time periods, time boundaries, time windows, time intervals, spacecraft, orbit, instruments, and also manages their DB interfaces.

Classes

- PIBoundary -- This class is used to describe boundaries for data acquisition and data processing within PGEs.
- PIDbIfManager -- This file defines a helper class to manage the various database interfaces.
- PIInterval -- This class provides methods for evaluating and manipulating time intervals based on start and stop times.
- PIOrbitBoundary -- The goal of this class is to encapsulate the knowledge of how to determine boundary times.
- PIOrbitModel -- The PIOrbitModel class is a table used by PIOrbitScheduled for calculating the estimated times for an orbit.
- PIOrbitPeriod -- The goal of this class is to store the platform of in a period such that the PITime class can perform Period & Boundary operations.
- PIPathModel -- This class will give us one to one mapping of our current absolute path number to instrument-specific path number.
- PIPeriod -- This class defines a period of time.
- PITime -- This class is a utility class for management of Time within the PDPS, it is derived from the standard Rogue Wave class but overrides it's behaviour to expect as a standard to be constructed from UTC, and to format to UTC, which is the standard that is
- PITimeService -- Helper class to return ensure that only one instance of the EcTiTimeService class is initialized, so that when an application runs in the DCE environment it can use DCE time.

Templates

- PObjectPool -- This class defines a pool for planning objects (PRs, DPRs, PGEs etc.) The pool is primarily to be used as a static member of the Object class.

library RequestProcessing

/ecs/formal/DM/GatewayCSCI/src/RequestProcessing

The RequestProcessing library provides the EcDmV0ToEcsGateway with support for processing requests from the EDG client.

It supports the following requests :

- (i) Directory Search (ii) Inventory Search (iii) Search by Granule ID (iv) Price Estimate Request (v) Integrated Browse (vi) FTP Browse (vii) Product Request

Classes

- **AcquireItem** -- This class combines the distribution parameters and the UR for each acquire to be submitted to the data server.
- **DmGwAcquireItem** -- This class combines the distribution parameters and the URs for each acquire request to be submitted to the data server.
- **DmGwAcquireRequest** -- This class supports basic product requests processing.
- **DmGwBrowseRequest** -- This Class contains all the information and the operations required to submit a Browse request to the data server.
- **DmGwDirectoryRequest** -- This class performs abort, submit, extract header, and build output information for the directory request.
- **DmGwErrorResult** -- This class performs build error and return error.
- **DmGwExternalAcquireRequest** -- This class extends **DmGwAcquireRequest** to support Billable product requests processing.
- **DmGwGranule** -- Class **DmGwGranule** encapsulates all necessary information to build GRANULE group for INVENTORY SEARCH RESULT.
- **DmGwInternalAcquireRequest** -- This class extends **DmGwAcquireRequest** and supports non-billable Product Acquire processing.
- **DmGwInvESDTRreference** -- This class performs get spatial extent, rectangle, point, polygon, start & end date, UR.
- **DmGwInvQuery** -- This class performs creating gateway inventory query object.
- **DmGwInvRequests** -- This class performs submit, abort, find data servers, extract search informations, build inventory queries, build search requests, and build the outputs.
- **DmGwInvSearchRequest** -- This class performs submit, abort, complete, failed, and connect data server.
- **DmGwODLLabelIO** -- **DmGwODLLabelIO** class provides threadsafe interface to read/write the ASCII ODL definitions from/to ODL tree.
- **DmGwPriceEstRequest** -- This is used in Price Estimate Request for Floating scene.
- **DmGwProductRequest** -- This class extracts the Product Request ODL message, processes the request and formulates the result ODL message for the client.
- **DmGwResultGroup** -- Class **DmGwResult** encapsulates all necessary information to build RESULT group for INVENTORY SEARCH RESULT.
- **DmGwScanSizeUtility** -- This class contains calculates the size in scans of the floating scene in a subinterval.
- **DmGwSearchRequest** -- This class contains all the information and the operations required to submit a search request to the data server for production history or ancillary data UR.
- **DmGwV0BrowseRequest** -- This class extract the Browse Request parameters from the ODL tree, submits the requests.
- **DmGwV0ECSMapper** -- This class maps the V0 types to ECS types and vice versa.
- **DmGwV0EcsRequestReceiver**

- DmGwV0ProductRequest -- This class is an interface for DmGwServerConnection to connect to DORRAN to send billable Product Requests.
- DmGwV0Request -- This is the base class for the request types that can be processed by V0 gateway.
- LineItem -- This class represents the V0 LINEITEM type in Product Request message.
- ServerAlloc -- This class is used for indicate its UUID and if the server is busy.

library srf

/ecs/formal/COMMON/CSCI_SRF/src/srf

The SRF library provides the necessary mechanism for the application to communicate asynchronously with another application.

Classes

- EcSrAsynchRequest_C -- This class represents the context of an SRF request currently in execution.
- EcSrAsynchRequest_S -- This is the server side representation of a request in progress.
- EcSrIDTime -- The only purpose of this class is to put the structure of reqID, the time when the request is put in the dictionary, and the timeout in to a list that allows the cleanupThread to index through it.
- EcSrRequestDispatcher -- This class performs request dispatching.
- EcSrRequestServer_C -- This class represents the clients view of an SRF server.
- EcSrRequestServer_S -- This is the server side representation of a client/server relationship.
- EcSrTransport_1_0 -- This is the transport server for SRF.
- EcSrTransport_1_0_ABS -- Abstract base class for EcSrTransport.
- EcSrTransport_1_0_Mgr -- Manager class for EcSrTransport.

Appendix C. Java Archives

This appendix lists all ECS Java Archives in alphabetical order. Each archive name is followed by a brief description of that archive. This is followed by a list of stored in the archive.

The first sentence of a class description is taken from the comments, if any, in the code for that class.

EcCIWbJdt

The Java DAR Tool is a java-based web application accessed through a web browser. The Java DAR Tool does not require the use of DCE. The Netscape Enterprise Server is used to serve the web pages for the Java DAR Tool and the ODFRM. The Java DAR Tool handles user requests to acquire ASTER data from the satellite. The user specifies when and where the data is to be taken. The DAR is submitted to the ASTER GDS in Japan. In addition to the submission and modification of acquisition requests, the Java DAR Tool allows users to search the ASTER DAR database (via MOJO and the DAR Comm. Gateway) in order to examine the status of DARs (i.e., the number and quality of acquired scenes). DARs can be searched on a parameter basis or by their DAR Ids.

Java Classes

- AbstractCellEditor -- This class encapsulates the housekeeping chores required by the TableCellEditor.
- AbstractDataHandler -- AbstractDataHandler processes the mouse events for an Area Of Interest or Area of Search point selection.
- AbstractDrawing -- This abstract class is used to draw point series as defined by the implementing class.
- AbstractMapModel -- AbstractMapModel is an abstract class that provides the default implementation for most of the methods in the IMapModel interface.
- AbstractMessage -- AbstractMessage is an abstract class that defines the generic data message.
- AbstractMessageAdapter -- Abstract adapter to deliver *relevant* message data to interested party.
- AbstractMessageManager -- The MessageManager is a messaging service.
- AbstractRenderer -- This abstract class renders an IDrawing object.
- AbstractSceneFormatter -- Copyright (C) 1998-1999, Raytheon Systems Company.
- AbstractSearchCriteriaFormatter
- AbstractSpinnerModel
- AbstractTreeModel -- This abstract class provides default implementations for most of the methods in the TreeModel interface.

- AbstractXARFormatter
- AbstractXARSearchResultFormatter
- AcquiredScenes -- Simple container class to hold scenes within a XAR that has returned after performing a search.
- AcquiredScenesDialog -- AcquiredScenesDialog display the acquired scenes panel.
- AcquiredScenesPanel -- AcquiredScenesPanel creates the Table with the ScenesDataModel.
- ActionCommand -- The data object for GUI specifications.
- ActionCommandList
- ActionsList -- ActionListEndOfActionList.
- Address -- This class provides the capability to specify the general request command to MOJO (such as JDTREQUEST, ADVERT, ADVERTDATA, etc.) and the JDT request type (such as SUBMITDAR, MODIFYDAR, etc.).
- AdvertisementRequest -- This class helps assemble the payload and command (in form of Packet) for advertisement request in the following format.
- AgentProxy
- Aoi
- AoiDrawing -- This class is used to draw the Aoi polygon on .
- AOIHandler -- AOIHanler processes the mouse event for the Area Of Interest point selection.
- AoiOverlay -- Overlay request for AOI.
- AoiRenderer -- This class renders an AOI (Area of Interest) component.
- AoiViewBuilder -- AoiViewBuilder responsible for construction and connection of Aoi Viewer's components, the major components in Aoi Viewer are spatial map and scenes table. Inorder to get a custom spatial map the builder provides information to the to build a custom
- AoiViewerDialog -- Aoi ViewerDialog displays spatial map screen and a list of available DAR ID.
- Aos
- AosDrawing -- This class is used to draw the Aos polygon on.
- AOSHandler -- AOSHandler processes mouse events for an Area Of Interest point selection.
- AosOverlay -- Overlay request for AOS.
- AosRenderer -- This class renders an AOS (Area of Search) component.
- ASCII_CharStream -- An implementation of interface CharStream, where the stream is assumed to contain only ASCII characters (without unicode processing).
- Attribute -- This class manages various aspects of an attribute - including its name, key, type, specifications, and so on.
- AttributeDataTypeException -- General purpose exception for attribute management in JDT.
- AttributeDictionary -- This class provides a capability to manage a set of attributes in an efficient manner.

- AttributeDictionaryBuilder -- This class reads the raw attributes from a file, parse the data to build an attribute, and then inserts the attribute in the attribute dictionary.
- AttributeException -- General purpose exception for attribute management in JDT.
- AttributeManager -- This class reads in attributes from a file (containing attributes in a tabular format) and builds an attribute dictionary by parsing various components of an attribute.
- AttributeManagerTest -- This class provides drivers for testing the attribute management by reading the attribute data file and building the attribute dictionary both in stand alone and in client/server mode.
- AttributeMessage
- AttributeNotFoundException
- AttributeParseException -- This exception is used to indicate any parsing problems while analyzing the attribute data after reading from a file.
- AttributeServlet -- A servlet to read attributes from a plain text file and then to create the attribute dictionary.
- AttributeSpecification -- This class provides the capability to store and manipulate various specifications for an attribute.
- AttributeSummaryScreen -- AttributeSummaryScreen displays and updates the XAR Attribute Selections and their values.
- AttributeValidationException -- General purpose exception for attribute management in JDT.
- BandComponent -- BandComponent is a wrapper class for selecting the bands.
- BoxRules
- BudgetRequestServlet -- Class description goes here.
- ButtonCellEditor -- This class enables a button to be pressed within a table and executes the action listeners and propagates the event from the button to the table.
- Capitalization -- This is a class having static methods to serve as utilities in development of other classes.
- Channel -- Channel is a topic channel.
- CheckboxButtonControls -- Checkbox Button Controls.
- CheckBoxCellEditor -- This class enables a checkbox to selected within a table and executes the action listeners and propagates the event from the button to the table.
- ColumnGroup
- ColumnHelper
- ColumnRange -- ColumnRange is used for setting the group of xar attributes' position in the table. The state of the display status is kept and the group can be either collapsed and expanded.
- ColumnSettings
- CommonProjection -- These methods were converted from GCTP (General Cartographic Transformation Package) of the USGS Abstraction of map projection.

- **CommunicationsManager** -- CommunicationsManager is the implementation of the network communication service.
- **ComponentConstants**
- **ComponentFactory** -- This class provides capabilities to create new component/object, and to setup help and other resources to the created component.
- **ConfigurationManager** -- This class encapsulates the JDT configuration parameters in one place.
- **ContextSensitiveHelpListener** -- Handles the context-sensitive help.
- **CoordConvert** -- Converts a coordinate from a pixel to a projection and from some projection to a pixel.
- **CopyDialog** -- This is a file dialog that allows the user to select a folder to place a Node into.
- **CoverageDetailsListener** -- Class description goes here.
- **CoverageDetailsScreen** -- This class enables the user to enter the Coverage Details for a Data Acquisition Request.
- **CoverageDetailsScreen** -- This class enables the user to enter the Coverage Details for a Data Acquisition Request Search.
- **CoverageDetailsSetting** -- This class contains the coverage details settings for a xAR data acquisition request.
- **CursorChanger** -- This class provides a convenient way of setting busy cursor on a component.
- **CustomHeaderRenderer** -- CustomHeaderRenderer renders the column header.
- **CustomLabel** -- Custom Label.
- **CustomMapRenderer** -- CustomMapRenderer is map data renderer.
- **CustomResultDataModel** -- CustomResultDataModel is the data model for both of the header and detail tables, it accesses the XAR and the XARStates objects to populate both tables.
- **DarBudgetAgent** -- Class description goes here.
- **DarIDComboBox** -- This class implements the combo box for the Search by ID screen.
- **DarIDComboBoxModel** -- This class implements the combo box data model for the Search by ID screen.
- **DarIDSearchScreen** -- DarIDSearchScreen presents and collects special search setting data from the user.
- **DarIDTable** -- This is the Dar ID table.
- **DarIDTableModel** -- This is the Dar ID table model.
- **DARIDUpdater** -- Watcher class that keeps track of status messages from the Submission Agent.
- **DataChangedEvent** -- The Data Change event is used to communicate changes in the data acquisition request data hierarchy.
- **DataChangedListener** -- This interface provides a means of notifying an implementing class when xAR data has been changed.

- DataInspector -- DataInspector takes the contents of a persistent object and displays the persistent attribute and value.
- DataReader
- DataWriter -- Creates a file for output and returns the BufferedWriter on demand.
- DateComponent -- The DateComponent class houses methods to create the component, add listeners and get and set the values of the month day and year.
- DateModel -- The DateModel class supplies the methods to set the bounds, get the months, days and years and the actions needed when the spinner is incremented or decremented.
- DateSpinner -- The DateSpinner class houses the methods to register listeners and the actions needed when focus is gained or lost in the numeric textfields.
- DateUtility -- The DateUtility class converts a String to a Calendar or a Calendar to a String.
- DayHour -- The DayHour class represents the data for a SpinnerModel containing only days and hours.
- DayHourComponent -- The DayHourComponent class houses methods to create the component, add listeners and get and set the values of the days and hours
- DayHourModel -- The DayHourModel class supplies the methods to set the bounds, get the days and hours and the actions needed when the spinner is incremented or decremented.
- DayHourSpinner -- The DayHourSpinner class houses the methods to register listeners and the actions needed when focus is gained or lost in the numeric textfields.
- DebugMessage -- Debug message used by the Debugger.
- DefaultSpinnerModel
- DefaultSpinnerRenderer -- .
- DeleteNodeListener -- Removes the current node.
- DialogBuilder -- ToolBarBuilder responsible of directing the ToolBarMgr to build and connect the toolbar, it provides the access to other components for the ToolBarMgr to accomplish the connecting process.
- DialogGeometry
- DialogManager -- Manages dialog instances.
- DictionaryNotFoundException
- DisplayBarMgr -- Manages the display bar.
- Distance
- DownloadAgentProxy
- Downloader -- Class description goes here.
- DownloadManager
- DownloadMessage
- DownloadSearchCriteriaServlet
- DownloadServlet
- DownloadXARSearchResultServlet

- DownloadXARServlet
- EditButton
- Editor -- This abstract class allows sub-classes to set modes for the screen.
- EmailXARFormatter
- ErrorConstants
- ErrorMessage -- Error Message used by the ErrorHandler.
- EventQueueManager
- FileOpenDialog
- Folder -- The Folder class is a representation of Composite of the composite design pattern (see specification for Node class for starting material).
- FolderSelectionListener -- The class is a listener for the drop-down Folder selection list on the XAROrganizerScreen.
- GainSettingsComponent -- GainSettingsComponent is a user interface component that displays a collection of BandComponents.
- GeneralScreen -- GeneralScreen displays the contents of the Primary Attribute screen and the Telescope screen.
- GeneralScreen -- GeneralScreen displays the contents of the Primary Attribute screen and the Telescope screen.
- GeneralUtilities
- GenericDialog -- Simple JDialog with main panel as a BorderLayout.
- GeographicalOverlayRenderer -- Makes the request to the mapRequester to fetch an image from the map server and receives the resulting image.
- GroupSettings
- GUIConstants
- Header
- HelpAgentProxy
- HelpClient -- Initializes and displays Java Help's Help Broker. Provides static help service..
- HelpConstants
- HelpData -- This is a data structure to hold the help related information, such as help id, tool tip, and instructional help text.
- HelpDictionary -- This class provides a capability to manage a set of help attributes in an efficient manner.
- HelpListener -- Handles the static help.
- HelpManager -- This class reads in the help data from a file (containing help information in a tabular format) and builds an help dictionary by parsing various components of an help data.
- HelpMessage -- Help Message generated by components that are help enabled and consumed by components that uses help.

- HelpServlet
- HidingCellRenderer
- HidingHeader
- HTMLFormatter
- HTMLSearchCriteriaFormatter
- HTMLXARFormatter
- HTMLXARSearchResultFormatter
- IAgentProxy -- INetworkAgentProxy is an interface to a NetworkAgentProxy.
- IAttribute
- IAttributeManager -- This is an interface for attribute management.
- IBuilder -- This is the interface class for all Spatial screen builders to implement, all implementors will know how to build, connect, layout, register and set Adaptor for all of their responsible components.
- ICommunicationsManager -- ICommunicationManager is the interface to the network communication service.
- IComponentFactory -- Copyright (c) 1998, Raytheon Company, its vendors, and suppliers.
- IComponentSupport -- IComponentSupport is an interface that allows Objects to specify that they want component support services like context-sensitive help, etc...
- IComponentUpdate -- This interface is for the implementor to handle the event generated from the map layer component generator.
- IConnection -- IConnection is a generic connection interface that allows the application to commit or rollback a request, open or close the connection, etc.
- IDataHandler
- IDictionary -- Provides a general purpose interface for dictionary behavior.
- IDictionaryElement -- Provides general behavior for elements needed to build a dictionary.
- IDrawing -- This class is used to draw the AOS polygon on.
- IGUIComponent -- This interface provides guidelines for performing certain activities on a GUI component while creating it or later retrieving attributes from it.
- IHelpManager -- This is an interface for the HelpManager.
- IImageIconFactory
- IImageLoadListener -- IImageLoadListener is an interface defined by the map services.
- IJDTManager -- Interface to the JDT Manager Fascade dispatcher.
- ILabelUpdate -- Interface to ensure that the implementations can update a label.
- ILayer -- ILayer is an interface to a layer in the map.
- ILayerManager -- ILayerManager is the interface to the layer management service.
- ILayerRenderer -- ILayerRenderer provides the extension to the rendering capabilities of the Map.

- ILoadImage -- ILoadImage interface will the methods responsible for notifying the thread group that a thread has started and stopped.
- ImageEvent -- ImageEvent is an event that is fired when an image is ready to be used.
- ImageIconConstants
- ImageIconFactory
- IMap -- Map interface provides interfaces to update and refresh layers.
- IMapManager -- Public interface to map manager.
- IMapScreenAdapter
- IMessage
- IMessageHandler -- The push consumer of messages.
- IMessageManager -- The IMessageManager provides a uniform interface to all messaging services.
- IMode -- IMode interface is implemented by components that permit mode changes.
- IModelManager -- IModelManager is an interface to the model services.
- INode
- INodeVisitor -- The INodeVisitor Interface allows the application to extend the functionality of the Node class without modifying the Node class structure.
- InstructionWindow
- InternalMessage -- Error Message used by the ErrorHandler.
- InvalidNodeException -- This exception can be thrown under circumstances when there is any problem in accessing a particular node (a folder or a leaf).
- IOrderedMap
- IOverlay -- Interface to all overlays; makes sure they can set and get the overlspecification and get and set an image.
- IPersistence -- IPersistence is interface to the persistent object, which allows the PersistenceManager to extract the attributes to write and read from persistence storage.
- IPersistenceManager -- IPersistenceManager is the interface to the persistence service.
- IRange -- This is a general purpose interface for implementing range objects.
- IRenderer
- IRequest
- IRequestRenderer
- IResultSet -- IResultSet is an interface to all types of result sets.
- IRowSetting
- IRowTableModel
- ISceneRegistry
- ISceneVisitor -- Copyright (c) 1998, Raytheon Company, its vendors, and suppliers.

- IScreen -- This interface allows the screens to synchronize their displays with the current data and layout their components.
- ISearchCriteria -- The ISearchCriteriaVisitor Interface allows the application to extend the functionality of the SearchCriteria class without modifying the SearchCriteria class structure.
- ISearchCriteriaVisitor -- The ISearchCriteriaVisitor Interface allows the application to extend the functionality of the SearchCriteria class without modifying the SearchCriteria class structure.
- IServiceProvider
- IUserProfileManager
- IValidationListener -- A VetoableChange event gets fired whenever a bean changes a "constrained" property.
- IXAR -- This is an interface for xAR class, defining all the methods needed for xAR objects.
- IXARManager -- The IXARManager is the interface to the data acquisition service.
- IXARSearchResult -- The IXARSearchResultVisitor Interface allows the application to extend the functionality of the XARSearchResult class without modifying the XARSearchResult class structure.
- IXARSearchResultVisitor -- The IXARSearchResultVisitor Interface allows the application to extend the functionality of the XARSearchResult class without modifying the XARSearchResult class structure.
- IXARVisitor -- The IXARVisitor Interface allows the application to extend the functionality of the XAR class without modifying the XAR class structure.
- JDTApplet -- This class is the entry point for the JDT Application.
- JDTConstants
- JDTEventDispatcher -- Monitors all user interface actions, processes the event, delegates the action to the Method Dispatcher.
- JDTEException -- JDTEException is an abstract class that represents the base exception class for the JAVA Data Acquisition Request Tool.
- JDTManager
- JDTRequirements -- Change the visible root in the File Manager.
- JDTScreen -- Navigation to the other screens, serves as main screen through which all others may be accessed.
- JDTTreeCellRenderer -- Interface to provide custom tree nodes that displays information like node name and time of creation.
- JDTTreeModel -- JDTTreeModel is a custom tree model which extends the AbstractTreeModel.
- JDTUtilities
- LabelCellEditor -- This class enables a button to pressed within a table and executes the action listeners and propagates the event from the button to the table.
- LabelComponent

- LabelTextComponent -- .
- Layer -- Layer is a graphical user interface container that represents a layer in the map.
- LayerManager -- LayerManager is the implementation of the layer management service.
- LayerModelEvent -- LayerModelModel event is used to communicate a change in the layer position.
- LoadImage -- Runnable thread for fetching an image.
- LogEntry
- LoginDialog
- LoginManager
- LookAngle -- LookAngle represents the angle off from nadir.
- LookAngleComponent -- LookAngleComponent displays and updates LookAngle.
- MailManager -- This class provides a capability to create a very simple text/plain message and to send it.
- Map -- Map is a user interface component that presents data in layered format.
- MapHandler -- Class for mouse event handling and conversion to coordinates.
- MapManager -- Map Manager manages the creation of the map by keeping track of overlays.
- MapModelException -- Signals that a MapModel exception has occurred.
- MapScreenAdapter
- MapScreenBuilder
- MapServerRequester -- Responsible for fetching the image from the server.
- MapUpdateThread
- MenuManager -- Dynamically changes the menu bar based on a context switch in the application.
- MessageDispatcher -- MessageDispatcher is a thread that monitors a message queue.
- MessageProcessor -- The MessageProcessor class provides a service available to developers and testers to aid them in code maintenance and testing.
- MessageProcessorTest -- This class provides driver for testing MessageProcessor.
- MessageQueue -- MessageQueue is a thread safe queue for messages.
- MethodDispatcher -- Manages actions that are triggered when a button press event occurs.
- MethodStore -- This class provides a storage location for reflection look ups.
- ModelConnection -- ModelConnection is a class that maintains a single connection to model.
- ModelManager -- The model manager maintains connections to model.
- ModelResultSet -- ModelResultSet is the result set that is return after executing the query.
- ModelResultSetIterator
- ModificationRequest -- This class assembles and handles the payload for dar modification requests.

- ModificationUpdater -- Watcher class that keeps track of status messages from the Submission Agent.
- ModifyCloudCoverageDialog
- ModifyXARFormatter
- MojoAgentProxy -- NetworkAgentProxy is a class that is created upon a request to send a message to a remote system on the network.
- MOJOMessage
- MojoResultSet -- MojoResultSet is the result set that encapsulates the Mojo response after a message has been sent to Mojo.
- MojoResultSetIterator -- The MojoResultSetIterator is a class that is used to simplify the interface for the MojoResultSet.
- MojoServlet -- MojoServlet is a NetworkAgent triggered by the NetworkAgentProxy.
- MOJOXARFormatter
- NameChangeEvent -- The Name Change event is used to communicate changes of node names within the data hierarchy.
- NameChangeListener -- This interface provides a means of notifying an implementing class when node name has been changed.
- NamedMessageManager -- Class description.
- NewDataEvent -- NewDataEvent is used to communication new data selection to the clients of the WorkSpace.
- NewNodeListener
- NewScreenEvent -- NewDataEvent is used to communication new data selection to the clients of the WorkSpace.
- Node -- Classes Node, Folder, xAR, and SearchCriteria represent the COMPOSITE design pattern.
- NodeLocator
- NodeRenderer -- Interface to provide a custom display for JDT nodes for use anywhere single nodes are displayed.
- NodeResultSet
- NodeSelectionKeyListener -- NodeSelectionKeyListener delegates the event(s) for selecting a node via the keyboard to the XAROrganizerScreen.
- NodeSelectionMouseListener -- NodeSelectionMouseListener delegates the event(s) for selecting a node on the XAROrganizerScreen's JTree. The trigger for this event is in the mouseClicked method..
- NodeWriter
- NumericTextField -- This class defines a text field and adds a key listener.
- OdfrmHTMLGenerator
- OnlineHelpComponent

- OrderedHash -- A class derived from HashMap which overcomes the problem where the order of the keys may change with each rehash.
- OutOfBoundsException -- General out of bounds exception.
- OverlayAdapter
- PanControl -- Controls panning on the screen.
- PanicMessage
- ParseException -- This exception is thrown when parse errors are encountered.
- ParseException -- This exception is thrown when parse errors are encountered.
- ParserUtils -- A collection of reusable routines for parsing the return data from MOJO.
- PathFinder -- PathFinder is a node visitor that generates a path for a selected node.
- PathNotFoundException -- This exception can be thrown under circumstances when there is any problem in accessing a particular node (a folder or a leaf).
- PayloadFormatter
- PchDrawing -- This class is used to draw the panning cross-hair.
- PCHHandler -- PCHHandler process the mouse event for the Panning Cross-Hair point selection.
- PchRenderer -- This class renders an PCH (Panning Cross-Hair) component.
- Persistence -- The Persistence class is abstract base class that allows other class to interface to persistence storage.
- PersistenceException -- Thrown to indicate a problem while either saving data to or restoring data from the persistence storage.
- PersistenceManager -- PersistenceManager is the implementation of the persistence service.
- PersistenceManagerSkel -- This class acts as a receiver for persistence requests and marshals these to the real persistence manager.
- PersistenceManagerStub -- Client/user side interface to the Persistence capability.
- PersistenceMessage
- PersistenceResultMessage
- PopupMenuListener -- PopupMenuListener listens for mouse events on the File Manager.
- PrimaryAttribute -- This class contains the primary attributes for a xAR data acquisition request.
- PrimaryAttributeListener -- Class description goes here.
- PrimaryAttributeScreen -- PrimaryAttributeScreen displays and updates the PrimaryAttribute.
- ProgressIcon -- A component that displays the correct icon based upon the status value.
- Properties
- Quadrant -- The quadrant class represents one scene in a xAR data acquisition result.
- Qualification

- QualificationList -- QualificationListQueryBeginServiceClassAttribute_LIB:001QueryEnd.
- Qualifier
- QualifiersList
- QueryResultsParser -- Parser to process the generic text format for MOJO XAR Status results data.
- QueryResultsParser -- Parser to process the generic text format for MOJO XAR Status results data.
- QueryResultsParserConstants
- QueryResultsParserTokenManager
- RadioButtonControls -- Radio Button Controls.
- Range -- This is a general purpose class representing a range of values of an attribute.
- RenameNodeListener -- Removes the current node.
- ResourceFactory
- ResourceHolder -- This class provides capabilities to access child components of the parent component supplied during instantiation.
- RestoreException -- Thrown when trying to restore a particular data and the process failed.
- ResultHeader
- ResultHeaderRenderer
- ResultList -- Simple container class to hold XARs that return after performing a search.
- ResultManager -- ResultManager manages the result display, it facades the request to the ResultTabManager, the ResultTableAdapter and the ResultTableFactory to construct and display the result sets.
- ResultMessage -- Results Message is used to carry the results of searches or submits.
- ResultRowVisitor -- ResultRowVisitor visits a xar, it extends AbstractXARFormatter to get a list of xar attributes, the visit method will map the order of xar attributes to the pre-fixed display arrangement for each attribute.
- ResultTableAdapter -- ResultTableAdapter handles the selection event for setting the selected xar id, sorting the rows by column and hiding/unhiding groups of columns.
- ResultTableBuilder -- ResultTableFactory constructs the customer table models and the tables with the adapters connected.
- ResultTablePanel -- ResulttablePanel connects the header table and the detail table together in a panel.
- ResultTabManager -- ResultTabManager manages the group of the result tabs either returned from the search or selected by the user from the organizer screen.
- Root -- Root is the top most node in the tree hierarchy.
- Row -- Row holds a list of result attributes in arranged order for displaying in the result table.
- Rubberband
- RubberbandRectangle

- RuleValidation -- The RuleValidation class houses all of the rules that pertain to the Data Acquisition Request.
- SaveException -- Thrown to indicate that saving data in persistence storage could not complete successfully.
- Scene -- This class represents one scene from a xar.
- SceneRowVisitor -- ResultRowVisitor visits a xar, it extends AbstractXARFormatter to get a list of xar attributes, the visit method will map the order of xar attributes to the pre-fixed display arrangement for each attribute.
- ScenesAdapter -- ScenesAdapter handles Scenes viewer's mouse event and scenes table's selection event.
- ScenesDataModel -- ScenesDataModel accesses the AcquiredScenes object to get the Vector of Scene, and gets each Scene's attributes..
- ScenesIDDataModel -- ScenesIDDataModel is a wrapper which gets sceneID and status from the ScenesDataModel.
- ScenesObservable
- ScenesParser -- Parser to process the SubXARStream contents returning through MOJO.
- ScenesRequest -- Following the pattern in this tool, we have a request for getting a chunk of scenes from GDS.
- SearchByAttributesRequest -- Following the pattern in this tool, we have a request for getting a list of XARs that satisfy a search criteria from GDS.
- SearchByAttributesUpdater -- Watcher class that keeps track of status messages from the Submission Agent.
- SearchByIdRequest -- Following the pattern in this tool, we have a request for getting a list of XARs based upon a list of IDs from GDS.
- SearchCriteria -- A SearchCriteria is a container object for performing a XAR Status query on GDS.
- SearchResultMessage -- Results Message is used to carry the results of searches or submits.
- SearchResultScreen
- SearchResultToolBarPanel -- SearchReultToolBar provides 4 functions.
- SearchScreen
- SearchStatusDialog -- This dialog contains the Search Status Panel.
- SearchStatusPanel -- This panel that contains the status table and toolbar for the status table.
- SearchStatusToolBar -- The tool bar for the search status screen.
- SearchSubmissionAgent -- The SearchSubmission agent controls the whole thread of sending a XAR Status search request through MOJO to GDS and routing the returned results.
- SearchSubmissionServlet
- SelectionTableAdapter -- Adapter for both updating the table and notifying the tables receiver of row changes in the table .
- ServiceSpecification

- ServletMessage
- ServletProxy
- SessionMessage
- SessionServlet
- SortDecorator
- SpatialComponentType -- Type safe way of implementing constants of the type of aoi screens.
- SpatialConstants
- SpatialScreen
- SpatialScreenBuilder -- This class is responsible of building the specific spatial screen, it directs all sub component builders to construct all required components and to plug all components together.
- SpatialScreenDirector -- This class uses the builder to assemble a spatial screen.
- SpatialScreenType -- Type safe way of implementing constants of the type of spatial screens.
- SpatialSetting -- This class contains the spatial settings for a xAR.
- SpatialThreadGroup -- SpatialThreadGroup manages threads that fetch requests for images from the map server or render themselves.
- SpecialRequestListener
- SpecialRequestScreen -- SpecialRequestScreen presents and collects special request setting data from the user.
- SpecialRequestScreen -- SpecialRequestScreen presents and collects SpecialRequest search setting data from the user.
- SpecialRequestSetting -- This class contains the special request settings for a xAR data acquisition request.
- Spinner -- Copyright (c) 1998, Raytheon Company, its vendors, and suppliers.
- SpinnerCheckBox
- SpinnerController -- SpinnerController - This class defines the control for the spinners.
- SpinnerModel
- SpinnerRenderer
- StatusMessage -- Status Message used to hold information on search and submit statuses.
- StatusRowData -- Contains the data from various classes that is needed to form a row in the table.
- StatusTable -- This class represents the search status table.
- StatusTableButtonRenderer -- Renders the buttons in the table.
- StatusTableCheckBoxRenderer -- Renders the checkbox's in the table..
- StatusTableLabelRenderer -- Renders the icons in the table.
- StatusTableModel -- The class represents the table model for the status table.

- `StatusTableMultilineRenderer` -- Renders the multiline column headers in the table.
- `StatusTableTextRenderer` -- Renders the icons in the table.
- `SubmissionAgent` -- This class acts as Submission manager for submitting both original requests as well as modification requests.
- `SubmissionRequest`
- `SubmissionServlet` -- Servlet to process DAR submit request.
- `SubmissionValidator` -- Submission validator performs mandatory checking on the data acquisition request (DAR).
- `SubmitException` -- Thrown during the xAR submission period.
- `SubmitRequestUpdateMessage` -- `SubmitRequestUpdates Message` is used to carry the results of searches or submits.
- `SubscriptionMessage` -- Results Message is used to carry the results of searches or submits.
- `SubscriptionRequest` -- This class process the payload for subscription, taking inputs from advertisement request.
- `SunAngle` -- `SunAngle` range represents the minimum and maximum sun angle range, the angle of the sun's ray incident to the point at which the pictures are taken..
- `SunAngleComponent` -- `SunAngleComponent` displays and updates the `SunAngle`.
- `SuspendActivateDialog`
- `SystemHelpManager` -- Manages system initiated help (e.g. Instruction Help and ToolTip Help..
- `TabManager`
- `TargetList`
- `TelescopeScreen` -- `TelescopeSettingScreen` allows the user to select telescope parameters.
- `TelescopeSetting` -- This class contains the telescope settings for the xAR data aquisition request.
- `TemporalListener` -- This class listens to the attributes for the Temporal Settings and when notified of a change in value, checks the new value to be sure that it does not violate any of the rules associated with the Temporal Settings.
- `TemporalScreen` -- `TemporalSettingScreen` collects `TemporalSetting` data.
- `TemporalScreen` -- `TemporalSettingScreen` collects `TemporalSetting` data.
- `TemporalSetting` -- This class contains the temporal settings for the xAR data aquisition request.
- `TextAreaDocument` -- This document model was created to filter whitespace characters (especially TAB and carriage returns) from the JDT text areas as they tend to cause parse errors in the MOJO proxies (as they should).
- `Token` -- Describes the input token stream.
- `TokenMgrError`

- ToolBarMgr -- Dynamically changes the menu bar based on a context switch in the application.
- TopicMessageManager
- TypeNotFoundException
- TypeUtilities
- UpdateButton
- UserProfileManager
- ValidationContext
- ValidationException -- Thrown to indicate that the validation of an attribute (a chunk of data) is not in its stable state.
- ValidationSupport -- This is a utility class that can be used by beans that support constrained properties.
- VAOIFormatter -- VASFormatter collects data from other sources to create VAS.
- VASFormatter -- VASFormatter collects data from other sources to create VAS.
- ViewingGeometryListener -- This class listens to the attributes for the Viewing Geometry Settings and when notified of a change in value, checks the new value to be sure that it does not violate any of the rules associated with the Viewing Geometry Settings.
- ViewingGeometryScreen -- ViewingGeometryScreen displays and updates the ViewingGeometrySetting.
- ViewingGeometryScreen -- ViewingGeometryScreen displays and updates the ViewingGeometrySetting.
- ViewingGeometrySetting -- This class contains the viewing geometry settings for a xAR data acquisition request.
- ViewSceneBuilder -- ViewSceneBuilder creates and connects components for scenes viewer dialog.
- ViewScenesDialog -- ViewScenesDialog is a dialog with spatial map screen and scene ID/Status table.
- WindowLevelHelpListener -- Handles the window-level help.
- Workspace
- WorkspaceListener
- XAR -- XAR class represents a leaf object in the composition (a leaf object has no children).
- XARContentParser -- Parser to process queryXARContents returning through MOJO.
- XARId -- Container to hold a XARId in a SearchCriteria for queryXARSummary.
- XARIdDataModel -- XARIdDataModel is a data model. it is used for displaying xar id within the xar id table.
- XARIdList -- Container to hold a list of XARIDs in a SearchCriteria for queryXARSummary.
- XARIdTableAdapter -- XARIdTableAdapter connects the xar id table to MapScreen.

- XARManager -- The XARManager is the implementation of the data acquisition service.
- XAROrganizerScreen -- XAROrganizerScreen is a object-oriented file system manager.
- XARScreen
- XARSearchResult -- Persistable container derived from Folder that contains search criteria and results.
- XARStatus -- This is the data object that is sent by the SearchSubmissionAgent to the client (ie. the SearchSubmissionUpdater).
- XARSummaryParser -- Parser to process queryXARSummary returning through MOJO.
- XARSummaryTreeCellRenderer -- Interface to provide custom tree nodes that displays information like node name and time of creation.
- XARSummaryTreeModel -- XARSummaryTreeModel is a custom tree model which implements the Swing TreeModel Interface.
- XARValidationVisitor

Abbreviations and Acronyms

A

ABC++	Document Generator used to provide class level detail
ACMHW	Access and Control Management Hardware (Configuration Item)
ADC	Affiliated Data Center (National Oceanic and Atmospheric Administration only)
ADSHW	Advertising Service Hardware (Configuration Item)
ADSRV	Advertising Service (Computer Software Configuration Item)
AGS	ASTER Ground System
AIT	Algorithm Integration and Test
AITHW	Algorithm Integration and Test Hardware (Configuration Item)
AI&T	Algorithm Integration and Test
AITTL	Algorithm Integration and Test Tools (Computer Software Configuration Item)
AM-1	See TERRA
AMASS	Archival Management and Storage System <i>Storage Management Software created by EMASS, which is the Raytheon Company, which in turn owns a part of Hughes Aircraft company, owns a wholly owned subsidiary of E-Systems Inc. E-Systems.</i> <i>For more info you can visit: http://www.emass.com</i>
AOI	Area of Interest
AOS	ASTER Operations Segment
AP	Algorithm Package
APC	Access/Process Coordinators
API	Application Program Interface
AQA	Algorithm Quality Assurance
AQAHW	Algorithm Quality Assurance Hardware (Configuration Item)
AR	Action Request

ARS	Action Request System (by Remedy)
AS	Administration Stations
ASCII	American Standard Code for Information Interchange
ASTER	Advanced Spaceborne Thermal Emission and Reflection Radiometer
ASTGW	ASTER Gateway CSCI
ATM	Asynchronous Transfer Mode

B

BCP	Bulk Copy Program
BDS	Bulk Data Server
BLM	Baseline Manager

C

CAD	Computer Aided Design
CCB	Change Control Board (Raytheon Convention) Configuration Control Board (NASA Convention)
CCDI	ClearCase DDTS Integration
CCR	Configuration Change Request
CDE	Common Desktop Environment
CDR	Critical Design Review
CDRL	Contract Data Requirements List
CD-ROM	Compact Disk - Read Only Memory
CDS	Cell Directory Service
CGI	Common Gateway Interface
CHUI	Character-based User Interface
CI	Configuration Item
CLS	Client Subsystem
CM	Configuration Management
CMI	Cryptographic Management Interface
CN	Change Notice

CO	Contracting Officer
COTS	Commercial Off the Shelf (Software or Hardware)
CPU	Central Processing Unit
CRM	Change Request Manager
CSC	Computer Software Component
CSCI	Computer Software Configuration Item
CSMS	Communications and Systems Management Segment (ECS)
CSS	Communications Subsystem

D

D3	High Performance Tape cartridge system (used in DSS STMGT CSCI)
DAA	Data Availability Acknowledgement
DAAC	Distributed Active Archive Center
DADS	Data Archive and Distribution System
DAN	Data Availability Notice
DAP	Delivered Algorithm Package
DAR	Data Acquisition Request
DAS	Dual Attached Station
DB	Database
DBMS	Database Management System
DCCI	Distributed Computing Configuration Item
DCE	Distributed Computing Environment (OSF)
DCN	Document Change Notice
DDA	Data Delivery Acknowledgment
DDICT	Data Dictionary (Computer Software Configuration Item)
DDIST	Data Distribution (Computer Software Configuration Item)
DDN	Data Delivery Notice
DDR	Detailed Design Review
DDT	DAAC Distribution Technician
DDTS	Distributed Defect Tracking System

DESKT	Desktop (Computer Software Configuration Item)
DEV	Custom Developed Code
DFS	Distributed File System
DID	Data Item Description
DIPHW	Distribution and Ingest Peripheral Hardware Configuration Item
DLL	Dynamic Link Library
DLT	Digital Linear Tape
DM	Data Management
DMGHW	Data Management Hardware (Configuration Item)
DMS	Data Management Subsystem
DNS	Domain Name Service
DOF	Distributed Object Framework
DORRAN	Distributed Ordering, Reporting, Researching, and Accounting Network (EDC)
DPR	Data Processing Request
DPRID	Data Processing Request Identifier
DPS	Data Processing Subsystem
DR	Data Repository
DRPHW	Data Repository Hardware (Configuration Item)
DSS	Data Server Subsystem
DTS	Distributed Time Service

E

EBnet	Earth Observing System Data and Information System Backbone Network
ECN	Engineering Change Notice
ECS	Earth Observing System Data and Information Core System
EDC	Earth Resource Observation System (EROS) Data Center
EDF	ECS Development Facility
EDN	Expedited Data Set Notification
EDOS	Earth Observing System Data and Operations System

EDR	Expedited Data Set Request
EDS	Expedited Data Set
EISA	Enhanced Industry Standard Architecture
E-mail	Electronic Mail (also Email, e-mail, and email)
EMASS	E-Systems Modular Automated Storage Systems
EOC	Earth Observing System Operations Center
EOS	Earth Observing System
EOSDIS	Earth Observing System Data and Information System
EROS	Earth Resource Observation System
ESDIS	Earth Science Data and Information System (GSFC Code 505)
ESDT	Earth Science Data Type
ESOD	Earth Science On-line Directory
ESRI	Environmental Systems Research Institute
ETM+	Enhanced Thematic Mapper Plus (Landsat 7)

F

FCAPS	Fault, Configuration, Accountability, Performance, and Security services
FDDI	Fiber Distributed Data Interface
FH	Fault Handling
FLDB	Fileset Location Database
F&PRS	Functional and Performance Requirements Specification
FSMS	File and Storage Management System
FTP	File Transfer Protocol
FTPD	File Transfer Protocol Daemon

G

GB	gigabyte (10^9)
GCDIS	Global Change Data and Information System
GCMD	Global Change Master Directory
GDPS	Ground Data Processing System

GDS	Ground Data System (ASTER)
GFE	Government Furnished Equipment
GSFC	GODDARD Space Flight Center (NASA facility and DAAC)
GSMS	Ground System Management Subsystem (ASTER)
GUI	Graphical User Interface

H

HDF	Hierarchical Data Format
HDF-EOS	an EOS proposed standard for a specialized HDF data format
HIPPI	High Performance Parallel Interface
HMI	Human Machine Interface
HP	Hewlett Packard
HPOV	Hewlett Packard Open View (COTS product)
HTML	HyperText Markup Language
HTTP	HyperText Transport Protocol
HWCI	Hardware Configuration Item

I

IAS	Image Assessment System
ICD	Interface Control Document
ICLHW	Ingest Client Hardware (Configuration Item)
ID	User Identification (or Identifier)
IDG	Infrastructure Development Group
IDL	Interactive Data Language
I/F	Interface
IHCI	Internetworking hardware configuration item
ILG	Infrastructure Library Group
ILM	Inventory, Logistics, Maintenance (ILM) Manager
IMS	Information Management System (ECS element name)
INGST	Ingest (Computer Software Configuration Item)

INS	Ingest Subsystem
INHCI	Internetworking HWCI
I/O	Input/Output
IOS	Interoperability Subsystem
IP	Internet Protocol International Partner
IRD	Interface Requirements Document
IRR	Incremental Release Review
ISO	International Standards Organization
ISS	Internetworking Subsystem
I&T	Integration and Test

J

JESS	Java Earth Science Server
JEST	Java Earth Science Tool
JPL	Jet Propulsion Laboratory (DAAC)

K

KFTP	Kerberos File Transfer Protocol
------	---------------------------------

L

L0 - L4	Level-0 through Level-4 data (ECS)
L0R	Landsat Reformatted Data
LAN	Local Area Network
Landsat 7	Land Remote-Sensing Satellite
LaRC	Langley Research Center (DAAC)
LFS	Local File System
LPGS	Landsat 7 Level-1 Product Generation System
LPS	Landsat-7 Processing System
LZ77	Lempel-Ziv coding

M

MB	Megabyte (10^6)
Mbps	Megabits Per Second
MCF	Metadata Configuration File
MCI	Management Software Configuration Item (Computer Software Configuration Item)
MHWCI	(System) Management (Subsystem) Hardware Configuration Item
MIB	Management Information Base
MISR	Multi-Imaging SpectroRadiometer
MLCI	Management Logistics Configuration Item (Computer Software Configuration Item)
MM	Mode Management
MMO	Mission Management Office
MMS	Mode Management Service
M&O	Maintenance and Operations (Staff)
MOC	Mission Operations Center
MODIS	Moderate-Resolution Imaging SpectroRadiometer
MOJO	Message Oriented middleware of JEST (JDT) Objects
MOPITT	Measurements of Pollution in the Troposphere
MP	Message Passing
MS	Mass Storage Microsoft
MSS	System Management Subsystem
MTMGW	Machine to Machine Gateway Server
MTPE	Mission to Planet Earth
N	
NASA	National Aeronautics and Space Administration
NCR	Non-conformance Report
NESDIS	National Environmental Satellite, Data, and Information Service (NOAA)

Netscape	Browser (for user registration and search engine) Mail component for e-mail transfers
NFS	Network File System
NIS	Network Information Service
NMC	National Meteorological Center (located at National Oceanic and Atmospheric Administration)
NMS	Network Management System (EBnet)
NNM	Network Node Manager
NNTP	Network News Transfer Protocol
NOAA	National Oceanic and Atmospheric Administration
NSI	National Aeronautics and Space Administration Science Internet
NSIDC	National Snow and Ice Data Center

O

ODL	Object Description Language
ODPR	On-Demand Production Request
OEM	Original Equipment Manufacturer
OODCE	Object Oriented DCE
OPS CON	Operations Concept
OS	Operating System
OSI	Open Systems Interconnect
OV	Open View (an Hewlett Packard product)
OVW	Open View for Windows

P

PAN	Production Acceptance Notification
PC	Personal Computer
PDPS	Planning and Data Processing Subsystems
PDR	Preliminary Design Review
PDRD	Product Delivery Record Discrepancy

PDS	Production Data Set
PF	Process Framework
PGE	Product Generation Executive
PLANG	Production PLANninG (Computer Software Configuration Item)
PLNHW	Planning Hardware (Configuration Item)
PLS	Planning Subsystem
PM	EOS Afternoon Equator Crossing Mission
PPDS	Planning and Processing Data Server
PRONG	(Data) PROcessiNG (Computer Software Configuration Item)
PRR	Preliminary Requirements Review
PWB	Planning WorkBench

Q

QA	Quality Assurance
QDS	Quick Look Data Set (See Expedited Data Set)

R

RAID	Redundant Array of Inexpensive Disks
RAM	Random Access Memory
RCS	Request Communications Support
RDBMS	Relational Database Management System
REL	Release
RFA	Remote File Access
RFC	Request For Comments
RIP	Routing Information Protocol
RMS	Request Management Services
ROSE	Request Oriented Scheduling Engine
RPC	Remote Procedure Call
RSC	Raytheon Systems Company
RTF	Rich Text Format

S

SAGE III	Stratospheric Aerosol and Gas Experiment III
SAS	Single Attached Station
SATAN	Security Administrator Tool for Analyzing Networks
SBSRV	Subscription Server (Computer Software Component)
SCF	Science Computing Facility
SCSI	Small Computer System Interface
SDF	Software Development Folder
SDP	Science Data Processing
SDPS	Science Data Processing Segment (ECS)
SDPTK	Science Data Processing Toolkit Science Data Processing Toolkit (Computer Software Configuration Item)
SDSRV	Science Data SeRVer (Computer Software Configuration Item)
SGL	Silicon Graphics, Inc.
SIPS	Science Investigator-Led Processing Systems
SMC	System Management Center
SMP	Symmetric Multi-processor
SMTP	Simple Mail Transport Protocol
SMUX	Simple Network Management Protocol Multiplexing
SNMP	Simple Network Management Protocol
SPARC	Single Processor Architecture
SPRHW	Science Processing Hardware (Configuration Item)
SQL	Structured Query Language
SQS	Spatial Query Server
SRF	Server Request Framework
SSAP	Science Software Archive Package
SSIT	Science Software Integration and Test
STK	StorageTek
STMGT	Storage Management (Computer Software Configuration Item)
Sybase	(ECS) COTS database management product

SYSLOG System Log

T

TAR Tape Archive

TCP Transmission Control Protocol

TCP/IP Transmission Control Protocol/Internet Protocol

TEC Tivoli Enterprise Console

TELNET Telecommunications Network

TERRA EOS AM Mission spacecraft 1, morning equator crossing spacecraft series
-- ASTER, MISR, MODIS and MOPITT instruments

TME Tivoli Management Environment

TMR Tivoli Managed Region Server

TT Trouble Ticket

U

UDP User Datagram Protocol

UR Universal Reference

URL Universal Resource Locator

UUID Universal Unique Identifier

V

V0 Version Zero

V0 GTWAY Version Zero Gateway

VT Virtual terminal

W

WAN Wide Area Network

WKBCH WorkBenCH (Computer Software Configuration Item)

WKSHW Working Storage Hardware Configuration Item

WRS Worldwide Reference System

WS	Working Storage
WWW	World Wide Web

X

xAR	x Acquisition Request (where x is any kind of or generic acquisition request)
XBDS	Bulk Data Service Protocol
XDR	External Data Representation
XFS	Extended File System
XVT	XVT Software, Inc.

This page intentionally left blank.